

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

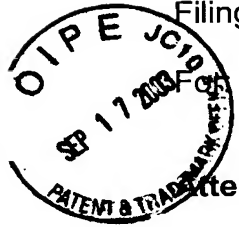
Applicants: Yasuhiro NAKAHARA

Serial No.: 10/606,227

Filing Date: June 26, 2003

METHOD AND APPARATUS FOR DERIVING COMPUTER SYSTEM  
CONFIGURATION

Attention: MS Missing Parts



**LETTER CLAIMING RIGHT OF PRIORITY**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

September 17, 2003

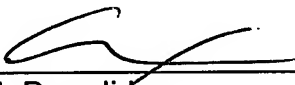
Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, applicants hereby claim  
the right of priority based on:

**Japanese Application No. 2002-186335**  
**Filed: June 26, 2002**

A Certified copy of said application document is attached hereto.

Respectfully submitted,

  
\_\_\_\_\_  
Carl I. Brundidge  
Registration No. 29,621  
ANTONELLI, TERRY, STOUT & KRAUS, LLP

CIB/jdc  
Enclosures  
703/312-6600

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日                      2 0 0 2 年    6 月 2 6 日  
Date of Application:

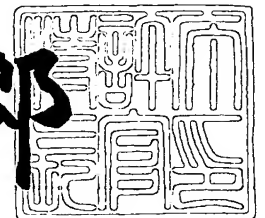
出 願 番 号                      特 願 2 0 0 2 - 1 8 6 3 3 5  
Application Number:  
[ST. 10/C]:                      [ J P 2 0 0 2 - 1 8 6 3 3 5 ]

出      願      人                      株式会社日立製作所  
Applicant(s):

2 0 0 3 年    7 月    9 日

特許庁長官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号    出証特 2 0 0 3 - 3 0 5 4 3 4 8

【書類名】 特許願

【整理番号】 KN1448

【提出日】 平成14年 6月26日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/60

【発明者】

    【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社  
                                 日立製作所 ソフトウェア事業部内

    【氏名】 中原 康博

【特許出願人】

    【識別番号】 000005108

    【氏名又は名称】 株式会社 日立製作所

【代理人】

    【識別番号】 100093492

    【弁理士】

    【氏名又は名称】 鈴木 市郎

    【電話番号】 03-3591-8550

【選任した代理人】

    【識別番号】 100078134

    【弁理士】

    【氏名又は名称】 武 顕次郎

【手数料の表示】

    【予納台帳番号】 113584

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンピュータのシステム構成導出方法及び装置

【特許請求の範囲】

【請求項1】 コンピュータシステムへの処理要求発生頻度とコンピュータのシステム構成とから処理要求を出してからその処理が終了するまでの応答時間を予測し、システム構成からその価格を計算し、与えられた応答時間Aと処理要求全体に対してその応答時間がA以上となる処理の確率として与えられた確率Bとに対して、確率がB以下となるシステム構成のうち、最も安価なシステム構成を導出することを特徴とするシステム構成導出方法。

【請求項2】 コンピュータシステムへの処理要求発生頻度とコンピュータのシステム構成とから処理要求を出してからその処理が終了するまでの応答時間を予測し、システム構成からその価格を計算し、与えられた応答時間Cと処理要求全体に対してその応答時間がC以下となる処理の確率として与えられた確率Dとに対して、確率がD以上となるシステム構成のうち、最も安価なシステム構成を導出することを特徴とするシステム構成導出方法。

【請求項3】 コンピュータシステム構成から応答時間を予測し、システム構成からその価格を計算し、処理要求を出してからその処理が終了するまでの応答時間として与えられた応答時間A、処理要求全体に対してその応答時間がA以上となる処理の確率として与えられた確率B及びジョブの処理要求の到着率Eに対して、応答時間がA以上である確率がB以下であるシステム構成のうち、最も安価なシステム構成を導出することを特徴とするシステム構成導出方法。

【請求項4】 コンピュータシステムへの処理要求発生頻度とコンピュータのシステム構成とから処理要求を出してからその処理が終了するまでの応答時間を予測し、システム構成からその価格を計算し、与えられた応答時間Aと処理要求全体に対してその応答時間がA以上となる処理の確率Bとに対して、確率がB以下となるシステム構成のうち、最も安価なシステム構成を導出し、導出された結果に基づいて、動的にシステム構成を変更することを特徴とするシステム構成導出方法。

【請求項5】 前記動的なシステム構成の変更は、予備のシステム構成要素

を起動あるいは停止させることにより行われることを特徴とする請求項4記載のシステム構成導出方法。

【請求項6】 コンピュータシステムへの処理要求発生頻度とコンピュータのシステム構成とから処理要求を出してからその処理が終了するまでの応答時間を予測する手段と、システム構成からその価格を計算する手段と、与えられた応答時間Aと、処理要求全体に対してその応答時間がA以上となる処理の確率として与えられた確率Bとに対して、応答時間がA以上である確率がB以下となるシステム構成のうち、最も安価なシステム構成を導出する手段とを備えたことを特徴とするシステム構成導出装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、コンピュータのシステム構成導出方法及び装置に係り、特に、トランザクション処理に対して一定の処理性能を保証するコンピュータのシステム構成を決定するコンピュータのシステム構成導出方法及び装置に関する。

【0002】

【従来の技術】

一般に、コンピュータシステムによるトランザクション処理において、システムに処理要求を出してから、その処理が終了するまでの時間（応答時間）が許容できる範囲にあることが必要とされることが多い。

【0003】

このため、処理の応答時間を予測するための理論として、例えば、待ち行列理論が知られており、この理論を用いて、平均サービス時間及び処理要求の到着率から平均応答時間を予測することがよく行われている。

【0004】

通常、コンピュータの性能設計（キャパシティプランニング）は、その技術分野の専門家が、経験に基づく知識及び待ち行列理論等の理論を利用して、要求される性能条件を満たすシステム構成を選び出して行われている。

【0005】

**【発明が解決しようとする課題】**

前述した従来技術による方法は、使用者が要求する一定の性能を保証するシステムを構成することは可能であっても、一定の性能を保証するシステムのうち価格が最小となるものを選び出すことが難しく、可能であっても膨大な時間と労力とを必要としてしまうという問題点を有している。

**【0 0 0 6】**

この理由は、一般に、一定の性能を実現するシステム構成の導出に当たって、CPUの処理性能、CPU数、主記憶容量、I/O処理速度等の組み合わせを考慮する必要があるが、この組み合わせが非常に多くあるためである。特に、一定の応答時間を保証する必要があるシステムの場合、応答時間が大きくなった場合のシステム性能改善対策は早急に行われる必要があり、短時間で最適システム構成を導出することが要求されるが、前述の従来技術は、このような要求に対応することができないという問題点を有している。

**【0 0 0 7】**

また、応答時間に条件を与えて、一定の性能を保証する場合、単位時間当りの要求到着数が平均到着数を中心として変動することを考慮すると、平均応答時間に制約を与えるよりも、応答時間が一定値以上になる確率に制約を与える方が適切である。この理由は、平均応答時間が小さい場合でも、許容できない応答時間が許容できない頻度で起こり得るからからである。前述した従来技術は、このような点について配慮されていない。

**【0 0 0 8】**

本発明の目的は、前述した従来技術の問題点を解決し、必要な処理性能を持つコンピュータのシステム構成の価格が最小となるものを自動的かつ短時間で決定することができるコンピュータのシステム構成導出方法及び装置を提供することにある。

**【0 0 0 9】****【課題を解決するための手段】**

本発明によれば前記目的は、コンピュータシステムへの処理要求発生頻度とコンピュータのシステム構成とから処理要求を出してからその処理が終了するまで

の応答時間を予測し、システム構成からその価格を計算し、与えられた応答時間 A と処理要求全体に対してその応答時間が A 以上となる処理の確率として与えられた確率 B とに対して、確率が B 以下となるシステム構成のうち、最も安価なシステム構成を導出することにより達成される。

#### 【0010】

すなわち、本発明は、待ち行列理論に基づく応答時間の予測方法を応用し、性能保証対象のコンピュータシステムで実行されるジョブの性質を表わすジョブデータ、処理要求の到着率、及び、そのシステムの CPU 性能、メモリ容量等のシステム構成要素のデータから処理要求の応答時間を予測する機能と、システム構成要素の価格テーブルからシステム構成の価格を計算する機能とを有し、前述の 2 つの機能を協働させることにより、「処理要求を出してからその処理が終了するまでの応答時間が A 以上となる処理の確率を B 以下にする」という指定方法で、一定の処理性能を保証し、かつ、その性能を実現するシステムの中で、価格が最小となるコンピュータシステムのシステム構成（最適システム構成）を、短時間で導出することを可能としている。

#### 【0011】

前述したような最適システム構成の導出は、応答時間が一定値以上である確率及びシステムの合計価格を、システム構成を決定するパラメタセットである、CPU 処理性能、CPU 数、メモリ容量等の関数としてとらえ、応答時間の値が一定値以上となるシステムパラメタ空間の部分空間、あるいは、部分集合の中から、価格関数が最小となるパラメタを決めることにより実現している。

#### 【0012】

##### 【発明の実施の形態】

以下、本発明によるコンピュータのシステム構成導出方法及び装置の実施形態を図面により詳細に説明する。

#### 【0013】

図 1 は本発明の第 1 の実施形態によるコンピュータのシステム構成導出装置を含む構成導出システムの構成を示すブロック図である。図 1 において、101 は構成導出装置、102 は構成導出部、103 は価格計算部、104 は応答時間計

算部、105は価格データ記憶装置、106はシステム構成記憶装置、107はジョブデータ記憶装置、108は到着率記憶装置、109は性能保証対象コンピュータ群、110は性能保証対象コンピュータ、111は性能測定装置、112はシステム構成データ、113は監視サーバ、114は入力装置、115は出力装置、116はネットワークである。

#### 【0014】

図1に示す構成導出システムは、構成導出装置101と、監視サーバ113と、1または複数の性能保証対象コンピュータ110による性能保証対象コンピュータ群109とがネットワーク116を介して接続されて構成される。構成導出装置101は、構成導出部102、価格データ記憶装置105、システム構成記憶装置106、ジョブデータ記憶装置107、及び、到着率記憶装置108を含んで構成され、構成導出部102は、価格計算部103及び応答時間計算部104を含んで構成される。また、構成導出装置101は、入力装置114及び出力装置115に結合されている。

#### 【0015】

性能保証対象コンピュータ群109を構成する各性能保証対象コンピュータ110は、トランザクション処理を行うコンピュータシステムであり、到着する処理要求に対してトランザクション処理を行っている。性能保証対象コンピュータ群109のそれぞれの性能保証対象コンピュータ110は、システム性能測定装置111とシステム構成データ112とを含んで構成されている。

#### 【0016】

性能保証対象コンピュータ110内の性能測定装置111は、ジョブ実行によって参照される仮想記憶方式における仮想ページの識別子（仮想ページ番号）を時間順に記録した情報（以下、仮想ページ参照経路データ）、ジョブ実行におけるCPU処理時間、ページング処理以外のI/O処理時間を測定している。また、システム構成データ112は、図6により後述するように、CPUの処理性能、CPU数、主記憶容量、I/O処理速度、及び、仮想ページ1ページ当たりのページング時間の情報を有している。

#### 【0017】

監視サーバ 113 は、ネットワーク 116 を介して接続されている各性能保証対象コンピュータ 110 から、そのシステムのシステム構成データ 112、性能測定装置 111 が出力する仮想ページ参照経路データ、1 回のジョブ実行に要する CPU 処理時間データ、I/O 処理時間データを収集し、そのデータを構成導出装置 101 に転送する。また、監視サーバ 113 は、一定の間隔で、各性能保証対象コンピュータへの単位時間当たりの処理要求数（到着率）を測定し、到着率データを構成導出装置 101 に転送する。

#### 【0018】

構成導出装置 101 は、一定の間隔で監視サーバ 113 から転送される到着率データを受信し、到着率記憶装置 108 に記憶させている。

#### 【0019】

構成導出部 102 は、価格計算部 103 と応答時間計算部 104 とを含み、価格計算部 103 の機能と応答時間計算部 104 の機能とを協働させることによって、最適なコンピュータのシステム構成を導出する。なお、構成導出部 102 の処理の詳細は図 3 に示すフローにより後述する。

#### 【0020】

図 7 は価格データ記憶装置 105 に格納されている価格データの構成を説明する図である。価格データ記憶装置 105 に格納される価格データは、システム構成要素の性能と対応する価格の情報であり、CPU 価格データ 701、主記憶装置価格データ 702、及び、補助記憶装置価格データ 703 の各テーブルから成る。CPU 価格データ 701 は、CPU の性能 704 とその価格 705 とにより構成され、主記憶装置価格データ 702 は、主記憶装置の容量 706 とその価格 707 とにより構成され、補助記憶装置価格データ 703 は、補助記憶装置の I/O 処理速度 708 とその価格 709 とにより構成される。

#### 【0021】

価格データ記憶装置 105 に格納されるシステム構成要素の性能とその価格とは、入力装置 114 から入力により、追加、削除、及び、変更することができる。また、説明している本発明の実施形態では、説明を簡単にするため、補助記憶容量は、全て十分な容量があるとし、容量による価格の違いを考慮しないことと

する。

#### 【0022】

図6はシステム構成記憶装置106に格納されているシステム構成データの構成を説明する図である。システム構成データは、各性能保証対象コンピュータ110毎にシステム構成記憶装置106に格納されている。

#### 【0023】

図6に示すコンピュータシステム1のシステム構成データ601は、CPU数602、CPU処理性能603、主記憶容量604、I/O処理速度605、及び、1ページ当たりのページング時間606により構成される。ページング時間は606は、仮想記憶方式の主記憶制御におけるページインまたはページアウトに要する1ページ当たりの平均時間で、ここでは、説明を簡単にするため、ページインとページアウトとに要する時間の違いを考慮しないこととする。

#### 【0024】

前述のシステム構成データ601は、監視サーバ113が各性能保証対象コンピュータ110から採取し、構成導出装置101に転送され、構成導出装置101がシステム構成記憶装置106に記憶している。

#### 【0025】

ジョブデータ記憶装置107は、各性能保証対象コンピュータ110で実行されるジョブに係るデータ（ジョブデータ）を記憶している。ジョブデータは、監視サーバ113が性能保証対象コンピュータ110から採取し、構成導出装置101に転送され、構成導出装置101が、ジョブデータ記憶装置107に記憶している。

#### 【0026】

図2はジョブデータ記憶装置107に格納されているジョブデータの構成を説明する図である。ジョブデータ201は、仮想ページ参照経路データ202とCPU・I/O処理量データ203とにより構成される。

#### 【0027】

CPU・I/O処理量データ203は、1回のジョブ実行に要するCPU処理時間とCPU処理速度との積であるCPU処理量204、及び、同時実行ジョブ

数が1の場合に、1回のジョブ実行におけるI/O処理時間とI/O処理速度との積であるI/O処理量205により構成される。

#### 【0028】

CPU・I/O処理量データ203は、図1におけるシステム構成記憶装置106が有するシステム構成データのCPU処理性能の値とI/O処理速度の値とを用い、また、監視サーバ113から受信したCPU・I/O処理時間データを用いることにより算出することができる。

#### 【0029】

仮想ページ参照経路データ202は、仮想記憶で参照した仮想ページ番号207を時間の経過に沿って参照順番号206に対応させて記録したデータ（仮想ページ参照経路データ）であり、参照された仮想ページについて、ページ内容の変更があったか否かを変更フラグ208で記録している。仮想ページ参照経路データ202とCPU・I/O処理量データ203とは、ジョブを特定する各ジョブ識別子のそれぞれについて、それぞれ1要素が設けられる。

#### 【0030】

構成導出部102は、指定した性能保証対象コンピュータ110において、指定したジョブのトランザクション処理について、処理要求を出してからその処理が終了するまでの応答時間がA以上となる処理の処理要求全体に対する確率がB以下であるシステム構成で、価格が最小となるものを導出する機能を有する。以後、記号Aをトランザクションの応答時間、記号Bを応答時間がA以上になる確率として用いる。

#### 【0031】

説明している本発明の実施形態は、トランザクション処理中のコンピュータシステムからジョブデータを採取して、目的の最適システム構成を導出している。しかしながら、構成導出装置102が機能するためには、入力としてのジョブデータさえあればよく、本発明は、既に稼動しているコンピュータシステムの性能を改善するためだけでなく、初めてシステムを構築する場合にも適用することができる。

#### 【0032】

図3は構成導出部102の処理動作を説明するフローチャートであり、次に、これについて説明する。

【0033】

(1) 構成導出部102は、始めに、入力装置114から、システム識別子、ジョブ識別子を読み込むと共に、性能保証対象109の各性能保証対象コンピュータ、例えば、性能保証対象コンピュータ110に対して設定される2つの値、すなわち、トランザクションの応答時間Aと、トランザクションの処理時間が応答時間Aを越える確率Bとを読み込む（ステップS301）。

【0034】

(2) 構成導出部102は、次に、性能保証対象のジョブに対応するジョブデータをジョブデータ記憶装置107から読み込む（ステップS302）。

【0035】

(3) 構成導出装置101は、監視サーバ113が各性能保証対象コンピュータ110から採取した構成導出装置101に転送するためのシステム構成データを読み込み、そのデータをシステム構成記憶装置106に記憶する（ステップS303）。

【0036】

(4) 構成導出部102は、図2に示すジョブデータの仮想ページ参照経路データを利用し、参照されたページが再度参照されるまでに参照される他の異なる仮想ページの数を変数とした、仮想ページ参照間隔確率分布を計算する。この計算方法について後述する（ステップS304）。

【0037】

(5) 構成導出装置101は、監視サーバ113が性能保証対象コンピュータ110から採取し構成導出装置101に転送するための性能保証対象コンピュータにおけるジョブ処理要求の発生率（到着率）を到着率記憶装置108から読み込む（ステップS305）。

【0038】

(6) そして、構成導出部102は、価格データ記憶装置107から読み出した価格データから価格関数を導出する。この導出方法については、後述する（ステ

ップ S 3 0 6)。

【0039】

(7) トランザクションの処理時間が応答時間 A を越える確率が確率 B 以下であるシステム構成で、価格が最小となるもの(最適解)を、ラグランジュの未定乗数法を使った数値解法により算出する。この最適解導出の詳細については、後述する(ステップ S 3 0 7)。

【0040】

(8) 構成導出部 1 0 2 は、ステップ S 3 0 7 の処理で最適解が見つかったか否かを判定し、見つからなかった場合、ここでの処理を終了する(ステップ S 3 0 8、S 3 1 1)。

【0041】

(9) ステップ S 3 0 8 の判定で、最適解が見つかった場合、応答時間が A 以上である確率を計算する(ステップ S 3 0 9)。

【0042】

(10) 最適化されたシステム構成パラメータセットとその価格及び応答時間が A 以上である確率を出力装置 1 1 5 に出力して、ここでの処理を終了する(ステップ S 3 1 0、S 3 1 1)。

【0043】

図 4 は図 3 のステップ S 3 0 4 での仮想ページ参照間隔確率分布計算部の処理動作を説明するフローチャートであり、次に、これについて説明する。ここで、仮想ページ参照間隔確率と呼んでいる確率は、英語では、“Distance Probability” と呼ばれているものである。後に詳述するが、図 8 のグラフ 8 0 3 は、参照間隔 8 0 2 と確率 8 0 1 との関係を表わした仮想ページ参照間隔確率分布の例である。

【0044】

(1) ジョブデータ記憶装置 1 0 7 に格納される仮想ページ参照経路データ(図 2 の 2 0 2)から、対応するジョブ識別子のデータを読み込む(ステップ S 4 0 1)。

【0045】

(2) 仮想ページ参照経路データを利用して、ページが参照されてから、最初に、再び同じページが参照されるまでに参照される、他の異なる仮想ページの数（参照間隔）を確率変数とする確率分布（仮想ページ参照間隔確率分布）を計算する（ステップ S 4 0 2）。

#### 【0 0 4 6】

(3) ステップ S 4 0 2 で算出した仮想ページ参照間隔確率分布を作業用記憶領域である一時記憶領域に記録する（ステップ S 4 0 3）。

#### 【0 0 4 7】

図 5 は図 3 のステップ 3 0 7 での最適解導出の処理動作を説明するフローチャートであり、次に、これについて説明する。

#### 【0 0 4 8】

(1) まず、変数“最低価格”に、設定できる最大の正の整数  $I_{MAX}$ （価格としてあり得ない大きな値）を設定する（ステップ S 5 0 1）。

#### 【0 0 4 9】

(2) CPU 数  $N_{CPU}$  を 1 から  $MAX_{CPU}$ （増設可能な最大 CPU 数）まで、1 ずつ増加させる。そして、 $N_{CPU}$  を設定するたびに、後述するステップ S 5 0 3 からステップ S 5 0 9 までの処理を実行する。 $N_{CPU}$  が  $MAX_{CPU}$  より大きくなったら、後述のステップ S 5 1 0 に移行する（S 5 0 2）。

#### 【0 0 5 0】

(3) 最適なシステム構成パラメータセットの最適解を計算する。この計算方法については後述する（ステップ S 5 0 3）。

#### 【0 0 5 1】

(4) ステップ S 5 0 3 の処理で最適解が見つかったか否かを判定し、最適解が見つかった場合、価格データ記憶装置 1 0 5 内の価格データを用いてシステム構成パラメータセットを補正する。すなわち、ステップ S 5 0 3 で得られたシステム構成パラメータセットに最も近い現実のパラメータセットを探索する。この探索方法については後述する。得られたパラメータセットは、仮システム構成パラメータセットとして設定される（ステップ S 5 0 4、S 5 0 5）。

#### 【0 0 5 2】

(5) ステップ S 5 0 5 で計算された、システムパラメータセットにおける価格を計算し、計算した価格が変数“最低価格”の値以下か否かを判定する（ステップ S 5 0 6、S 5 0 7）。

#### 【 0 0 5 3 】

(6) ステップ S 5 0 7 の判定で、計算した価格が変数“最低価格”の値以下であった場合、ステップ S 5 0 5 で計算した価格を変数“最低価格”に設定し、そのときの仮システム構成パラメータセットを最適システム構成パラメータセットとして設定する（ステップ S 5 0 8）。

#### 【 0 0 5 4 】

(7) ステップ S 5 0 4 で最適解が見つからなかった場合、ステップ S 5 0 7 で計算した価格が変数“最低価格”の値以下でなかった場合、また、ステップ S 5 0 8 の処理の後、CPU 数  $N_{CPU}$  を 1 から  $MAX_{CPU}$  までとして、全ての処理が終了したときに処理を終了する（ステップ S 5 0 9、S 5 1 0）。

#### 【 0 0 5 5 】

前述した処理において、変数“最低価格”の値が  $I_{MAX}$  に一致していれば、最適解が存在していないことを意味する。

#### 【 0 0 5 6 】

次に、応答時間が A 以上である確率 ( $P[t > A]$ ) を計算する処理動作について説明する。こことでは、説明を明確にするため、ユーザが利用できる主記憶領域でワーキングセットにない記憶領域は、全て利用可能フレーム（アベイラブルフレーム）群として、OS によって管理されているものとする。

#### 【 0 0 5 7 】

ワーキングセットは、各ジョブに固定的に割り当てられる主記憶領域である。ワーキングセットの大きさの設定及び変更は、例えば、一般的な汎用コンピュータ OS を用いることにより容易に実施することができる。ページインは、参照するページがワーキングセットに無く、ページリクレイムができない場合に実行される。ページリクレイム機能は、仮想記憶方式のオペレーティングシステム（OS）が持つ機能として一般的である。

#### 【 0 0 5 8 】

また、本発明の実施形態において、ジョブ実行におけるCPU使用時間及びI/O処理時間のデータは、ジョブが実行される際に、性能測定装置が測定することによって得ているが、必ずしも実際にジョブ実行が行われている必要はなく、ジョブに用いる実行プログラムからCPU処理量及びI/O処理量を予測することも可能である。

## 【0059】

応答時間がA以上である確率（ $P[t > A]$ ）は、次の2つのステップで計算することができる。

## 【0060】

ステップ1 平均サービス時間の予測計算

要求された処理が始まってから終わるまでの時間（サービス時間）の確率分布は、数1として示す式（1）の $1/\mu_{N_J}$ を平均サービス時間とする指数サービスとして求められる。

## 【0061】

【数1】

$$\frac{1}{\mu_{N_J}} = T_{CPU} + T_{I/O, N_J} + T_{Paging, N_J} = T_{N_J} \quad \dots (1)$$

## 【0062】

式（1）において、 $\mu_{N_J}$ は、ジョブ数 $N_J$ のジョブが同時実行される場合において、1つのCPUが単位時間あたりに処理する処理要求数（サービス率）である。また、 $T_{CPU}$ は、対象ジョブを1回だけ処理するために必要とされるCPU処理時間である。CPUの処理性能が $S_{old}$ から、 $S$ になることにより、 $T_{CPU}$ は、数2として示す式（2）のように $S_{old}S^{-1}T_{CPU}$ に変更される。

## 【0063】

【数2】

$$T_{CPU} \rightarrow S_{old} S^{-1} T_{CPU} \quad \dots (2)$$

## 【0064】

また、式（1）において、 $T_{I/O, N_J}$ は、ジョブ数 $N_J$ のジョブが同時に処理さ

れている場合において、I/O待ちによるジョブ実行1回当たりのCPUの待ち時間である。I/O待ち時間 $T_{I/O, N_J}$ は、ページングによるI/O時間を含まない。さらに、 $T_{I/O, N_J}$ は、同時実行ジョブ数が1である場合の1回のジョブ実行におけるI/O処理率 $R_{I/O}$ を示す数3として示す式(3)、及び、式(3)から求められる数4として示す式(4)から予測計算することができる。

【0065】

【数3】

$$R_{I/O} = \frac{T_{I/O}(\text{ページングを除いたI/O処理時間})}{T_{CPU}(\text{CPU処理時間}) + T_{I/O}(\text{ページングを除いたI/O処理時間})}$$

ただし、

$$\text{CPU処理時間: } T_{CPU} = \frac{A_{CPU}(\text{CPU処理量})}{S(\text{CPU処理速度})}$$

$$\text{I/O処理時間: } T_{I/O} = \frac{A_{I/O}(\text{I/O処理量})}{I(\text{I/O処理速度})}, \quad \dots (3)$$

【数4】

$$\frac{T_{CPU}}{T_{CPU} + T_{I/O, N_J}} = 1 - (R_{I/O})^{N_J}$$

あるいは、

$$T_{I/O, N_J} = \frac{(R_{I/O})^{N_J} \cdot T_{CPU}}{1 - (R_{I/O})^{N_J}} \quad \dots (4)$$

【0066】

I/O構成の変更により、I/O処理速度が $I_{old}$ から $I$ に変更されることによって、 $T_{I/O}$ は、数5として示す式(5)のように、 $I_{old} I^{-1} T_{I/O}$ に変更される。

【0067】

【数 5】

$$T_{I/O} \rightarrow I_{old} I^{-1} T_{I/O} \quad \dots (5)$$

【0068】

さらに、前述の式（1）において、 $T_{\text{paging}, N_J}$  は、ジョブ数  $N_J$  のジョブが同時に処理されている場合のページングによる I/O 時間である。この計算手順については後に詳述する。

【0069】

ステップ 2 応答時間が A 以上である確率（ $P[t > A]$ ）の計算

説明している本発明の実施形態は、1つの CPU が、マルチプログラミングによって、ジョブ数  $N_J$  の複数ジョブを同時実行することができる。そして、ここでは、待ち行列理論を適用するため、CPU 数  $N_{\text{CPU}}$ 、同時実行可能なジョブ数  $N_J$  のシステムを、CPU 数  $N_{\text{CPU}}$ 、サービス率  $\mu_{N_J}$  のシステムとみなして予測計算する。

【0070】

実際には、同時実行可能なジョブ数  $N_J$  が 1 より大きく、実行中を含めた待ち要求数が同時実行可能なジョブ数  $N_J$  より小さい場合、サービス率は、待ち処理要求数に依存して変化するため、時間的に一定ではない。

【0071】

説明している実施形態では、同時実行可能なジョブ数  $N_J$  を、後に説明する最適化によって決定するため、実際のサービス率は時間的に一定ではなくても、一定値  $\mu_{N_J}$  であるとして近似する。

【0072】

待ち行列理論により、CPU 数が  $N_{\text{CPU}}$ 、同時実行可能なジョブ数が  $N_J$  のシステムにおいて、処理中のものを含めた、処理待ち要求数が  $k$  である確率  $P_k$  は、数 6 として示す式（6）で表わされる。

【0073】

【数 6】

$$P_k = \frac{(a_{N_J})^k}{k!} P_0 \quad (0 \leq k \leq N_{CPU}),$$

$$P_k = \tilde{P}_k = \frac{(N_{CPU})^{N_{CPU}} (\rho_{N_J})^k}{N_{CPU}!} P_0 \quad (N_{CPU} < k)$$

$$P_0 = \left( \sum_{k=0}^{N_{CPU}-1} \frac{(a_{N_J})^k}{k!} + \frac{(a_{N_J})^{N_{CPU}}}{(N_{CPU}-1)!(N_{CPU}-a_{N_J})} \right)^{-1},$$

ただし  $a_{N_J} = \frac{\lambda}{\mu_{N_J}}, \rho_{N_J} = \frac{\lambda}{N_{CPU} \mu_{N_J}} \dots (6)$

【0074】

従って、CPU数が $N_{CPU}$ 、同時実行可能なジョブ数が $N_J$ のシステムにおいて、待ち時間が $T$ 以上の確率は、数7として示す式(7)で表わされる。

【0075】

【数 7】

$$P_q[t > T] = \sum_{k=1}^{\infty} \left\{ (\text{系内に} k \text{個の要求がある確率}) \times \left( \sum_{s=0}^{k-1} (\text{T時間内に} s \text{の要求処理が終了する確率}) \right) \right\}$$

$$= \sum_{k=1}^{\infty} \left\{ P_k \left( \sum_{s=0}^{k-1} \frac{(N_{CPU} \mu_{N_J} T)^s}{s!} e^{-N_{CPU} \mu_{N_J} T} \right) \right\}$$

$$= \sum_{s=0}^{N_{CPU}-1} \sum_{k=s+1}^{N_{CPU}} \frac{\lambda^k \cdot (\mu_{N_J})^{s-k} \cdot (N_{CPU} T)^s}{k! \cdot s!} \cdot P_0 e^{-N_{CPU} \mu_{N_J} T}$$

$$+ \frac{(N_{CPU})^{N_{CPU}}}{N_{CPU}!} \frac{1}{N_{CPU} \cdot \mu_{N_J} - 1} \cdot P_0 e^{-(N_{CPU} \cdot \mu_{N_J} - 1) \cdot T} \dots (7)$$

【0076】

応答時間は、待ち時間を用いて、

応答時間＝待ち時間＋実行時間、

で計算されるが、実行時間を平均実行時間  $1/\mu_{N_J}$  で置き換えることにより、応答時間が $A$ 以上である確率は、数8として示す式(8)から計算することができる。

【0077】

【数8】

$$P[t > A] = P_q[t > (A - \mu_{N_j}^{-1})] \quad \dots (8)$$

【0078】

次に、仮想ページ参照経路データからページングに要する時間を予測計算する手順について説明する。

【0079】

説明している本発明の実施形態は、性能保証対象コンピュータの主記憶管理が仮想記憶方式によるものとし、スワッピングは考慮せず、ページングできないページ（ページ固定ページ）は無いものとする。また、性能保証対象コンピュータで実行される複数のジョブは、同じプログラムによるもので、ジョブ実行の結果は、入力情報にのみ依存し、各ジョブ処理において、仮想ページ参照間隔確率分布が大きくは異なるものとしている。

【0080】

仮想ページ参照間隔確率分布が各ジョブで大きく異なる場合への拡張は、本発明の実施形態では示していないが、各ジョブの仮想ページ参照間隔確率分布を平均した確率分布を用いて、実施形態に適用させることも可能である。参照される仮想ページに対応する実ページがワーキングセットにあれば、ページイン処理の必要はない。また、参照される仮想ページに対応する実ページがワーキングセットに無い場合でも、オペレーティングシステムが管理するアベイラブルフレームに、必要なページの内容が変更されることなく残っていれば、ページリクレイム機能によって、そのフレームが再利用され、ページイン処理は行われない。

【0081】

いま、あるページが参照されると、その後に他のページが参照されても、参照されたページは、直ぐにはワーキングセットから削除されない。ワーキングセットに空きが無くなると、アベイラブルフレームキューから利用可能な1ページが取得され、ワーキングセットの中で直前の参照からの間隔が最も大きい（LRU）フレームがアベイラブルフレームキューにエンキューされ、ワーキングセット

から削除される。

#### 【 0 0 8 2 】

ページアウトは、ページリクレイム以外の目的で、アベイラブルフレームからフレームを取得する際に、そのフレームの内容を補助記憶装置に書き出す必要がある場合にのみ行われる。ここで、「書き出す必要がある」とは、①対応する補助記憶装置上のフレームがあり、かつ、その内容が主記憶上のフレームの内容と異なっていること、あるいは、②対応するフレームが補助記憶上にないが、再度参照する必要があることである。

#### 【 0 0 8 3 】

前述したようなことをふまえ、仮想ページ参照間隔確率と呼ばれる確率を用いることにより、ジョブ実行中にページングが必要となるページ数を予測計算し、ページングに要する時間を計算することができる。ここで、ページリクレイムによるフレームの再利用は、ページングに含めていない。

#### 【 0 0 8 4 】

図 2 に示す仮想ページ参照経路データ 2 0 2 の参照順番号 2 0 6 の  $i$  番目の仮想ページを参照してから、最初に、再びそのページが参照されるまでの間に参照される他の異なる仮想ページの数  $d$  の場合、対応する仮想ページ参照間隔  $D(i)$  は  $D(i) \equiv d$  で定義され、再度参照されることがない場合、 $D(i) \equiv \infty$  で定義される。

#### 【 0 0 8 5 】

仮想ページ参照間隔が  $d$  である確率  $P(d)$  は、母集団  $M = \{D(i) \mid i \in \text{すべての仮想ページ参照経路順序番号}\}$  から、数 9 として示す式 (9) により定義される。

#### 【 0 0 8 6 】

【数 9】

$$P(d) = \frac{\text{母集団 } M \text{ 内の要素 } d \text{ の数}}{\text{母集団 } M \text{ の総数}} \quad \dots (9)$$

#### 【 0 0 8 7 】

図 8 は仮想ページ参照間隔確率の確率分布の例を示す図である。この例に示す

ように、一般に、「参照の局所性」と呼ばれる性質により、一度参照したページが、短時間のうちに、再び参照される確率は高い。

#### 【0088】

本発明の実施形態は、各CPUが独立した主記憶装置を保有しており、各CPU上で独立してOSが稼動していることを想定している。従って、性能保証対象コンピュータは、ネットワーク装置等により相互に接続された、同じシステム構成を有する複数のシステム群と置き換えてもよい。

#### 【0089】

1つのCPUに対して、その全メモリ容量MからOS等の基本ソフトウェアに要するメモリ容量 $M_{fix}$ を減じたメモリ容量 $M_{user}$ （以下、利用可能メモリ容量）を仮想ページ1ページ当りのメモリ容量で割った値（ページ数）を $d_{user}$ とし、ワーキングセットのページ数を $d_{working\ set}$ とする。このとき、 $d_{user}$ は、数10として示す式（10）により表される。

#### 【0090】

【数10】

$$d_{user} = \frac{M - M_{fix}}{B \text{ (単位フレームの主記憶容量)}} = \frac{M_{user}}{B} \quad \dots (10)$$

#### 【0091】

複数のCPUが1つの主記憶装置を共有し、1つのOSのみが稼動している密結合マルチプロセッサシステムの場合、CPU1つ当たりのメモリ容量をM、CPU1つあたりの基本ソフトウェア用メモリ容量を $M_{fix}$ として換算するとよい。

#### 【0092】

図8において、区間804は、利用可能な全ページ数 $d_{user}$ を表わし、区間805は、ワーキングセットのページ数 $d_{working\ set}$ を表わし、区間806は、全アベイラブルフレーム数を同時実行されるジョブ数 $N_J$ で割った値、

$(d_{user} - N_J d_{working\ set}) / N_J$ 、  
を表わしている。

#### 【0093】

本発明の実施形態は、ページアウト処理が、アベイラブルフレームをリクレイム以外の目的で利用する場合で、そのフレームの内容が変更されており、補助記憶装置に書き込む必要が起きた場合にのみ行われる。

#### 【0094】

従って、あるページが参照されてから、次に再度参照されるまでの参照間隔が  $d_{user} + (d_{user} - N_J d_{working\ set}) / N_J$  ( $= d_{user} / N_J$ ) 以下であれば、あるいは、この参照間隔が図8の区間807の間隔以下であれば、その再参照の際にも、ページ内容が実記憶上に残っているため、ページインは行われない。

#### 【0095】

CPU数が1で、同時実行ジョブ数が1の場合、利用可能メモリの全てが対象ジョブに使用されるため、ページイン確率は、対象ページが主記憶上にない確率として、数11として示す式(11)で表わされる。

#### 【0096】

【数11】

$$P_{page\ in} = 1 - \sum_{d=1}^{d_{user}} P(d) \quad \dots (11)$$

#### 【0097】

また、ページアウト確率は、ジョブ処理において、ページ内容に変更がある確率を  $P_{change}$  として、数12として示す式(12)で表わされる。

#### 【0098】

【数12】

$$P_{page\ out} = P_{change} P_{page\ in} \quad \dots (12)$$

#### 【0099】

$P_{change}$  は、仮想ページ参照経路データ204のページ内容変更フラグ208がONである確率として、数13として示す式(13)で計算する。

#### 【0100】

## 【数 13】

$$P_{change} = \frac{\text{参照経路データにおける変更フラグ'ON'の総数}}{\text{仮想ページ参照経路データの長さ}} \quad \dots (13)$$

## 【0101】

この確率は、ページの内容が変更されてから、最初に再び参照されるまでに参照される他の異なる仮想ページの数を変数とした確率分布を計算することにより、より正確に計算することも可能である。

## 【0102】

ジョブ数  $N_J$  のジョブが同時に実行される場合、ワーキングセットのページ数を  $d_{working\ set}$  とし、ワーキングセット以外で利用可能なページが利用可能フレーム群として確保されているとすると、ページイン確率は、数 14 として示す式 (14) で予測計算することができる。

## 【0103】

## 【数 14】

$$P_{page\ in} = 1 - \frac{d_{user}}{\sum_{d=0}^{N_J} P(d)} \quad \dots (14)$$

## 【0104】

最適なシステム構成をラグランジュの未定乗数法で導出する場合、前述の確率は、次の数 15 として示す式 (15) ような確率密度の積分形式で表現する。

## 【0105】

## 【数 15】

$$P_{page\ in} = 1 - \int_0^{d_{user}/N_J} \hat{P}(m) dm \quad \dots (15)$$

## 【0106】

但し、非積分関数は、 $P([m])$  で定義される。 $[m]$  は実数  $m$  を超えない最大の整数である。

## 【0107】

また、ページアウト発生確率は、数 16 として示す式 (16) で予測計算する

ことができる。

【0 1 0 8】

【数 1 6】

$$P_{page\ out} = P_{change} P_{page\ in} \quad . . . (16)$$

【0 1 0 9】

1 回のジョブ実行において、ページングに費やされる時間は、1 ページのページインに要する時間と 1 ページのページアウトに要する時間とを、それぞれ  $t_{page\ in}$ 、 $t_{page\ out}$  として、数 1 7 として示す式 (1 7) で計算する。

【0 1 1 0】

【数 1 7】

$$T_{paging} = L_1 (t_{page\ in} \cdot P_{page\ in} + t_{page\ out} \cdot P_{page\ out}) + L_2 \cdot t_{page\ in} \quad . . . (17)$$

【0 1 1 1】

式 (1 7) において、最後の項は、参照される仮想ページへの、最初のページの読み込みに要する時間である。 $L_1$  は、仮想ページ参照経路データの長さ（仮想ページ参照順の最後の番号）であり、 $L_2$  は、主に、補助記憶装置から主記憶領域に読み込まれるジョブ実行用プログラムのための仮想ページの数である。

【0 1 1 2】

本発明の実施形態は、 $L_2$  が仮想ページ参照経路データにおける異なる仮想ページの総数から、最初の参照で、変更フラグが ‘ON’ である参照の数を減じた値として計算される。

【0 1 1 3】

ここでは、最初の参照において、変更フラグが ‘ON’ である仮想ページが、ジョブ実行用プログラム内で動的に確保されたページであり、最初の参照において ‘OFF’ である仮想ページがプログラムを格納するためのページであるとして概算している。

【0 1 1 4】

本発明の実施形態は、 $t_{page\ in}$  が  $t_{page\ out}$  に等しいとして、数 1 8 として

示す式(18)でページング時間を予測する。

【0115】

【数18】

$$T_{\text{paging}} = L_1 \cdot t_{\text{paging}} \cdot P_{\text{page in}} (1 + P_{\text{change}}) + L_2 \cdot t_{\text{paging}}$$

$$\text{ただし, } P_{\text{paging}} = P_{\text{page in}} = P_{\text{page out}}, \quad t_{\text{paging}} = t_{\text{page in}} = t_{\text{page out}} \dots (18)$$

【0116】

また、本発明の実施形態は、ページがワーキングセットにある場合に比べて、ページがワーキングセットになく、ページリクレイム可能な場合のページリクレイム処理による遅延時間はないとしている。

【0117】

ページリクレイムによる遅延時間が無視できない場合でも、1ページ当たりのページリクレイムに要する時間を測定することにより、ジョブ実行において、ページリクレイムに要する時間を予測計算することが可能である。

【0118】

次に、応答時間がA以上である確率がB以下である条件で、価格が最小となるシステム構成（最適解）を選び出すための方法について詳細に説明する。

【0119】

応答時間がA以上である確率がB以下である条件で、価格が最小となるシステム構成を選び出すことは、ある値Cについて、システム構成データ（ $s_1, s_2, \dots, s_n$ ）で、性能を表わす関数、性能関数  $F = F(s_1, s_2, \dots, s_n)$  が制約条件  $F \leq C$  を満たし、かつ、目的関数としての価格関数  $G = G(s_1, s_2, \dots, s_n)$  を最小にするシステム構成を探すことと同値である。

【0120】

本発明の実施形態における、システム構成パラメータは、（CPU処理性能S、CPU数 $N_{\text{CPU}}$ 、全メモリ容量M、I/O処理速度I、同時実行可能ジョブ数 $N_J$ ）であり、制約条件  $F \leq C$  は、応答時間がA以上である確率がB以下である条件に対応している。応答時間がA以上である確率がB以下である条件は、数19として示す式(19)に示すものとなる。

【0 1 2 1】

【数 1 9】

$$P[t > A] = P_q[t > (A - \mu_{N_j}^{-1})] \leq B \quad \dots (19)$$

【0 1 2 2】

後述するように、最適解であるための条件により、同時実行可能ジョブ数  $N_j$  は、他のシステム変数に独立な変数ではない。

【0 1 2 3】

価格を表わす関数、価格関数は、離散的な価格データを再現できる関数であるとし、数 2 0 として示す式 (2 0) により表されるものとする。

【0 1 2 4】

【数 2 0】

$$G = g_{CPU}(S, N_{CPU}) + g_{memory}(N_{CPU}M) + g_{I/O}(I), \quad \dots (20)$$

【0 1 2 5】

説明を簡単にするため、CPU 数  $N_{CPU}$  個の CPU 価格は、CPU 1 個の価格の  $N_{CPU}$  倍であるとする、CPU の価格関数  $g_{CPU}$  は、数 2 1 として示す式 (2 1) により表される。

【0 1 2 6】

【数 2 1】

$$g_{CPU}(S, N_{CPU}) = N_{CPU} \cdot g_{CPU}(C, 1) \equiv N_{CPU} \cdot g_{CPU}(C) \quad \dots (21)$$

【0 1 2 7】

CPU の価格関数  $g_{CPU}$ 、主記憶装置の価格関数  $g_{memory}$  及び補助記憶装置の価格関数  $g_{I/O}$  は、パラメータフィッティング法等により求めることができる。

【0 1 2 8】

図 9 は性能と価格との関係を表すグラフの例を示す図である。前述の各価格関数は、例えば、図 9 の 9 0 1 に示すように、(価格、性能) の分布の近似曲線として求められる。関数形は、例えば、数 2 2 として示す式 (2 2) のように、1

変数連続関数の和で表現される。

【0 1 2 9】

【数 2 2】

$$g(s) = g + g s + g s^2 + \dots \quad \dots (22)$$

【0 1 3 0】

近似曲線の適切な関数形（2 次式、3 次式、対数関数等）は、各システム構成要素の価格関数に依存している。

【0 1 3 1】

関数の決定には、不定性があっても、微分式  $g'(s)$  が条件  $g'(s) \geq 0$  ( $s > 0$ ) を満たすならば、最適なシステム構成の結果には大きくは影響しないため、例えば、図 9 の 9 0 2 のような折れ線で表わされる関数であってもよい。

【0 1 3 2】

価格関数は、一般に、システム構成変数 ( $s_1, s_2, \dots, s_n$ ) の各変数について増加関数であるため、式 (1 9) を満たし、かつ、価格を最低にするシステム構成を導出するための制約条件は、数 2 3 として示す式 (2 3) のように置き換えれば充分である（制約条件①）。

【0 1 3 3】

【数 2 3】

$$P_q \left[ t > (A - \mu_{N_J}^{-1}) \right] = B \quad \dots (23)$$

【0 1 3 4】

システム構成パラメータ、（CPU 処理性能  $S$ 、CPU 数  $N_{CPU}$ 、全メモリ容量  $M$ 、I/O 処理速度  $I$ 、同時実行可能ジョブ数  $N_J$ ）はジョブ数  $N_J$  を除いて独立である。

【0 1 3 5】

同時実行ジョブ数  $N_J$  は、確率  $P_q [t > (A - \mu_{N_J}^{-1})]$  を最大にする  $N_J$  として決定される。すなわち、与えられたシステム変数 ( $S, N_{CPU}, M, I$ ) に対して、 $N_J$  は、数 2 4 として示す式 (2 4) の解として決まる。

【0136】

【数24】

$$\frac{d}{dN_J} P_q \left[ t > (A - \mu_{N_J}^{-1}) \right] = 0 \quad \dots (24)$$

【0137】

前述において、式(23)の左辺に現れる値 $N_J$ は、式(24)の解としての $N_J$ の値である(制約条件②)。

【0138】

最適システムを求めるために、ラグランジュの未定乗数法を用いて、制約条件①及び制約条件②のもとで、目的関数としての価格関数 $G$ が最小となるシステム構成を選び出す。

【0139】

CPU数 $N_{CPU}$ の値を固定したときの最適システム構成パラメータセット( $S$ 、 $M$ 、 $I$ )は、ラグランジュの未定乗数 $\lambda_1$ 及び $\lambda_2$ を導入した数25として示す式(25)の関数 $U$ について、これから得られる(4+2)個の数26として示す式(26)を解くことによって、関数 $U$ の極値を計算して得られる。

【0140】

【数25】

$$U = G(s_1, s_2, \dots, s_n) + \lambda_1 \left( B - P_q \left[ t > (A - \mu_{N_J}^{-1}) \right] \right) + \lambda_2 \left( \frac{\partial}{\partial N_J} P_q \left[ t > (A - \mu_{N_J}^{-1}) \right] \right) \quad \dots (25)$$

【数26】

$$\partial U / \partial s_i = 0 \quad (s_i = S, M, I, N_J, \lambda_1, \lambda_2), \quad \dots (26)$$

【0141】

前述の極値解におけるシステム構成パラメータセット( $S$ 、 $M$ 、 $I$ )に対応するシステム構成が、与えられたCPU数が $N_{CPU}$ であるときの、最適化されたシ

ステム構成である。

#### 【0 1 4 2】

C P U 数を含めてシステム構成を決定するためには、C P U 数  $N_{CPU}$  を 1 から増設可能な最大 C P U 数  $MAX_{CPU}$  まで 1 ずつ増加させ、各 C P U 数における最適化されたシステム構成を導出し、その中から最低価格のシステム構成パラメータセット ( $N_{CPU}$ 、S、M、I) を決定する。

#### 【0 1 4 3】

前述の式 (2 6) に示す方程式の解は、非線形計画法等でよく用いられるニュートン法による逐次近似法を利用した数値から計算することができる。

#### 【0 1 4 4】

ニュートン法では、先の関数  $U$  について、次の数 2 7 として示す式 (2 7) のようにシステム構成パラメータ  $s^C = (s_i^C)$  の周りでテイラー展開する。

#### 【0 1 4 5】

##### 【数 2 7】

$$U(s^C + \Delta s) = U(s^C) + \sum \frac{\partial U}{\partial s_i} \Big|_{s=s^C} \Delta s_i + \sum \sum \frac{\partial^2 U}{\partial s_i \partial s_j} \Big|_{s=s^C} \Delta s_i \Delta s_j \dots (27)$$

#### 【0 1 4 6】

前述の  $s^C$  近傍で最小値を計算するためには、先ず、次の数 2 8 として示す式 (2 8) のベクトル  $V$  と行列  $Q$  とを計算する。

#### 【0 1 4 7】

##### 【数 2 8】

$$V^C = \left( \frac{\partial U}{\partial s_i} \Big|_{s=s^C} \right), \quad Q^C = \left( \frac{\partial^2 U}{\partial s_i \partial s_j} \Big|_{s=s^C} \right) \dots (28)$$

#### 【0 1 4 8】

式 (2 7) において、関数  $U$  を最小にする  $\Delta s$  は、数 2 9 として示す式 (2 9) で表される。

#### 【0 1 4 9】

【数 2 9】

$$\Delta s = (-1) \cdot (Q^C)^{-1} \cdot V^C \quad \dots (29)$$

【0 1 5 0】

目的の最小値を計算するには、 $s^C$  に  $(s^C + \Delta s^C)$  の値を設定し、テイラー展開の最小値解を計算する処理を繰り返し、解が収束するまで、つまり、 $\Delta s$  の大きさが零、あるいは、零とみなすことができるまで繰り返す。

【0 1 5 1】

式 (2 6) 2 6 の解 (最適システム構成パラメータセット) は、離散的な値を有する選択可能なシステム構成パラメータセットとは異なるため、求められた連続的な実数で表現された最適システム構成パラメータセットに最も近い現実の使用可能なシステム構成パラメータセットを選ぶ必要がある。

【0 1 5 2】

このため、図 5 により説明した最適解導出処理におけるステップ S 5 0 5 の処理でのシステム構成補正の処理は、計算で得られたシステム構成パラメータセット  $(s_1, s_2, \dots, s_n)$  と、実際に価格データに存在しているシステム構成パラメータセット  $(s'_1, s'_2, \dots, s'_n)$  との距離が最も近い現実のシステム構成パラメータセットを探索する。

【0 1 5 3】

このため、本発明の実施形態は、各システム変数  $s_i$  ( $i = 1, 2, \dots, n$ ) について、 $s'_i \geq s_i$  であって、 $s'_i - s_i$  が最小となるシステム構成データ  $s'_i$  を価格データの中から探すことにより最適化された現実のシステム構成パラメータセット  $(s'_1, s'_2, \dots, s'_n)$  を決定している。

【0 1 5 4】

例えば、図 9 において、最適解が 9 0 3 の座標で表わされる場合、9 0 4 で表わされるシステム構成要素が最適化されたシステム構成要素となる。

【0 1 5 5】

この最適化されたシステム構成の導出方法は、応答時間が A 以上である確率が

B以下である最低価格のシステムを構築するために、そのシステム構成要素を開発する場合に、開発する各システム構成要素の性能値を決定する方法として利用することができる。

#### 【0156】

説明している実施形態は、システム構成としての複数のCPUについて、各CPUが同じ処理性能を持つことを前提としている。処理性能が異なる複数のCPU環境において、待ち行列理論で応答時間を予測することは難しいが、この場合でも、シミュレーションによる計算手法を用いることにより、応答時間を予測することが可能である。

#### 【0157】

次に、本発明の第2の実施形態として、価格データを利用し、システムパラメータの全ての組み合わせについて、前述した式(19)の条件を満たすか否かを判定し、条件を満たす組み合わせの中から、価格が最低となる組み合わせを探索することにより最適解を得る例について説明する。この第2の実施形態は、システム構成要素の組み合わせの数が少ない場合に特に有効である。なお、第2の実施形態のシステム構成は図1に示すものと同一であってよい。

#### 【0158】

図10、図11は本発明の第2の実施形態での構成導出部の処理動作を説明するフローチャートであり、次に、これについて説明する。

#### 【0159】

を用いて説明する。

#### 【0160】

(1) 構成導出部102は、始めに、入力装置114から、システム識別子、ジョブ識別子を読み込むと共に、性能保証対象109の各性能保証対象コンピュータ、例えば、性能保証対象コンピュータ110に対して設定される2つの値、すなわち、トランザクションの応答時間Aと、トランザクションの処理時間が応答時間Aを越える確率Bとを読み込む(ステップS1001)。

#### 【0161】

(2) 構成導出部102は、次に、性能保証対象のジョブに対応するジョブデー

タをジョブデータ記憶装置 107 から読み込む (ステップ S1002)。

【0162】

(3) 構成導出装置 101 は、監視サーバ 113 が各性能保証対象コンピュータ 110 から採取した構成導出装置 101 に転送するためのシステム構成データを読み込み、そのデータをシステム構成記憶装置 106 に記憶する (ステップ S1003)。

【0163】

(4) 構成導出部 102 は、図 2 に示すジョブデータの仮想ページ参照経路データを利用し、参照されたページが再度参照されるまでに参照される他の異なる仮想ページの数を経率分布とした、仮想ページ参照間隔確率分布を計算する (ステップ S1004)。

【0164】

(5) 構成導出装置 101 は、監視サーバ 113 が性能保証対象コンピュータ 110 から採取し構成導出装置 101 に転送するための性能保証対象コンピュータにおけるジョブ処理要求の発生率 (到着率) を読み込む (ステップ S1005)。

【0165】

(6) 次に、変数「最低価格」に、値  $I_{MAX}$  を代入する。値  $I_{MAX}$  は、コンピュータ上で表現できる正の最大数とする (ステップ S1006)。

【0166】

(7) 全てのシステム構成パラメータセットの組み合わせについて、応答時間が A 以上である確率が B 以下であるか否かを判定するため、価格データに含まれる各システム構成要素の価格データから構成されるシステム構成パラメータセットで、判定していないもの (仮システム構成) を選び出す (ステップ S1007)。

【0167】

(8) ステップ S1007 で選び出した仮システム構成において、応答時間が A 以上である確率を最小にする同時実行ジョブ数とその確率とを計算する (ステップ S1008)。

**【0168】**

(9) 式(7)及び式(8)を使用して応答時間がA以上である確率を計算する(ステップS1009)。

**【0169】**

(10) ステップ1009で計算された確率がB以下であるか否かを判定し、計算された確率がB以下でない場合、後述するステップS1014に移行する(ステップS1010)。

**【0170】**

(11) ステップS1010の判定で、計算された確率がB以下であった場合、価格データから、各システム構成要素の価格を読み込み、価格の合計額を計算する(ステップ1011)。

**【0171】**

(12) ステップS1011において計算された合計額が変数「最低価格」の値より小さいか否かを判定し、計算された合計額が変数「最低価格」の値より大きければ、後述するステップS1014に移行する(ステップS1012)。

**【0172】**

(13) ステップS1013の判定で、計算された合計額が変数「最低価格」の値より小さければ、変数「最低価格」にステップS1011で計算された価格を設定し、対応する仮システム構成パラメータセットをシステム構成パラメータセットとして設定する(ステップS1013)。

**【0173】**

(14) 価格データから構成可能な全てのシステム構成パラメータセットが試されたか否かを判定し、全てが試されていない場合は、次の仮システム構成パラメータセットを決定するためにステップS1007からの処理を繰り返し、全てが試されたならばここでの処理を終了する(ステップS1014、S1015)。

**【0174】**

前述した処理において、変数「価格」がI<sub>MAX</sub>でないならば、性能条件を満たし、価格を最低にするシステム構成が存在しており、対応するシステム構成パラメータセットが「システム構成」に設定される。

## 【0175】

また、前述のステップS1011における価格の計算手順には、次の2つの方法がある。すなわち、①全く新たにシステムを導入する場合の価格計算、及び、②既にあるシステムを拡張して利用する場合の価格計算の2つである。②の場合、例えば、仮システム構成のCPU数が3、主記憶容量が2GBで、既存のシステムのCPU数が2で主記憶容量が1GBであるとする、価格の計算では、CPUについては1(=3-2)個、主記憶容量については、1(=2-1)GBの価格を計算ればよい。

## 【0176】

図12は図11におけるステップS1008での同時実行ジョブ数計算の処理を説明するフローチャートであり、次に、これについて説明する。

## 【0177】

- (1) 同時実行ジョブ数の変数 $N_J$ に初期値として1を設定する(ステップS1201)。
- (2) 後に詳述する $P[t > (A - \mu N_J^{-1})]$ の同時実行ジョブ数 $N_J$ における $N_J$ についての微分係数を計算する(ステップS1202)。
- (3)  $P[t > (A - \mu N_J^{-1})]$ の同時実行ジョブ数 $N_J$ における、 $N_J$ についての2回微分の微分係数を計算する(ステップS1203)。
- (4)  $\Delta N_J$ を計算する。 $\Delta N_J$ はステップS1202で算出された微分係数 $F_1$ 、ステップS1203で算出された微分係数 $F_2$ を使用して、 $\Delta N_J = -(F_1 / F_2)$ で計算される(ステップS1204)。
- (5) ステップS1204で算出された値 $\Delta N_J$ を変数 $N_J$ に加算し、 $N_J$ を更新する(ステップS1205)。
- (6) 収束を判定し、解が収束していれば、同時実行ジョブ数計算処理を終了する。 $\Delta N_J$ の大きさ $|\Delta N_J|$ が0、あるいは、0に充分近ければ収束していると判定する。ここで、「0に充分近い」とは、例えば、 $|\Delta N_J| / |N_J| < 10^{-4}$ を満たすことである(ステップS1206)。

## 【0178】

次に、本発明の第3、第4の実施形態として、構成導出装置が出力する最適シ

システム構成データを利用して、応答時間がA以上である確率がB以下であるように、システム構成を動的に変更する方法及びシステムの例について説明する。

#### 【0179】

図13は性能保証対象コンピュータが予備システム構成要素を有する場合における本発明の第3の実施形態の構成を示すブロック図である。図13において、1305は性能保証対象コンピュータ、1306は性能保証対象コンピュータ群、1307は動的システム構成変更装置、1308は予備システム構成要素であり、他の符号は図1の場合と同一である。

#### 【0180】

図13に示す本発明の第3の実施形態は、性能保証対象コンピュータ群1306の各性能保証対象コンピュータ1305が、動的システム構成変更装置1307と予備システム構成要素1308とを備えて構成されている点で図1に示した第1の実施形態の場合と相違している。

#### 【0181】

予備システム構成要素1308は、性能保証対象のジョブ処理を行っていないシステム構成要素である。また、動的システム構成変更装置1307は、要求があった場合に、予備システム構成要素を性能保証対象のジョブ処理を実行するためにシステム構成要素に追加する。

#### 【0182】

前述のように、必要に応じて、物理的に接続されているシステム構成要素を使用可能な状態または使用不可能な状態に切り替える機能は、“キャパシティオンデマンド”や“キャパシティリザーブ”として知られている。

#### 【0183】

構成導出装置101は、最適システム構成データを、性能保証対象コンピュータの動的システム構成変更装置1307に送信する。動的システム構成変更装置1307は、最適システム構成データを受信すると、そのシステム構成を実現するため予備システム構成をジョブ処理実行のために稼働させる。予備システム構成要素は、1つまたは2つ以上のCPU、あるいは、主記憶装置等である。

#### 【0184】

本発明の第3の実施形態は、使用するCPU数、使用する主記憶容量、及び、その使用時間等に応じて課金されることを想定している。そして、動的システム構成変更装置1307は、予備のシステム構成要素を起動あるいは停止させることにより、システムを停止させることなく、性能保証対象コンピュータのCPU数、主記憶容量等を変更することができる。

#### 【0185】

この本発明の第3の実施形態は、構成導出装置101が、出力装置115から最適なシステム構成情報を出力する代わりに、ネットワーク116を利用して、性能保証対象コンピュータが有する動的システム構成変更装置1307に最適なシステム構成情報を送信する。

#### 【0186】

動的システム構成変更装置1307は、システム構成導出装置101が送信した最適システム構成の情報を入力し、その情報に対応するようにシステム構成を動的に変更させている。

#### 【0187】

このため、本発明の第3の実施形態は、応答時間がA以上である確率がB以下となるようにシステム構成を動的に変更しつつ、これに要する課金額を最低に抑えることが可能である。

#### 【0188】

図14は1つの性能保証対象コンピュータが、複数のサブシステムからなる場合の本発明の第4の実施形態の構成を示すブロック図である。図14において、1403は構成変更指示部、1407は性能保証対象コンピュータ、1408はトランザクション分配部、1409～1411はサブシステム1～3、1412は出力装置、1413は入力装置、1414は処理結果記憶装置であり、他の符号は図1の場合と同一である。

#### 【0189】

図14に示す本発明の第4の実施形態は、構成導出装置101が、構成導出部102に加えて、構成変更指示部1403を備えて構成されている点、及び、性能保証対象コンピュータ1407が、トランザクション分配部1408、サブシ

ステム 1409～1411、出力装置 1412、入力装置 1413、処理結果記憶装置 1414 を備えて構成されている点で図 1 に示した第 1 の実施形態の場合と相違している。

#### 【0190】

各サブシステムは、性能保証対象のジョブ処理のために利用されるものと、それ以外の用途に利用されているものに分類される。ここでは、例えば、サブシステム 1409 は、性能保証対象のジョブ処理を行っており、サブシステム 1410、1411 は性能保証対象のジョブ処理を行っておらず、サブシステム 1410 は、ジョブ処理以外の用途にも使用されてなく、サブシステム 1411 は、ジョブ処理以外の用途に利用されているものとする。

#### 【0191】

サブシステム 1409～1411 によるサブシステム群は、物理的に独立したコンピュータシステム群で、ネットワーク機能によって互いに結合されているものであってもよい。この時、サブシステム群の全てのサブシステムについて、使用者（使用する権利を持った者）が同じでなくてもよい。また、説明している本発明の第 4 の実施形態における各サブシステムは、同等のシステム構成であり、同等の処理能力を有しているものとする。

#### 【0192】

本発明の第 4 の実施形態における構成導出部は、「応答時間計算部」の機能を利用して、要求される性能を実現するために必要なサブシステム数を計算する。この場合、サブシステム数を 1 ずつ増やすことにより、「応答時間計算部」の機能を利用して、応答時間が A 以上である確率が B 以下であるという条件を満たす最初のサブシステム数を算出する。応答時間を計算する処理を、目標の性能に達するまで繰り返せばよいので、前述した第 1 の実施形態に比べて処理が容易である。

#### 【0193】

以下に説明する本発明の第 4 の実施形態において、ジョブ処理以外の目的で利用されているサブシステムを予備サブシステムと呼ぶことにする。また、予備サブシステムの使用者は、性能保証対象のジョブを行っている者と異なっている。

よい。予備サブシステムは、そのサブシステムがジョブ処理用に使用される際に、性能保証対象のジョブを行っている者に課金、あるいは、所有者への送金を要求するサブシステム（有料サブシステム）と要求しないサブシステム（無料サブシステム）とに分類されている。サブシステムが無料サブシステムであるか、有料サブシステムであるかは、構成導出装置が有するサブシステムテーブルで調べることができる。

#### 【0194】

図20はサブシステムテーブルの構成を示す図である。サブシステムは、サブシステム識別子2001で識別されている。このテーブルは、各サブシステムについて、使用中であるか否かを知ることができる使用中／非使用中フラグ2002と、単位時間使用料情報2003とを有している。

#### 【0195】

図15～図17は本発明の第4の実施形態におけるシステム変更指示部の処理動作を説明するフローチャートであり、次に、これについて説明する。

#### 【0196】

(1) 構成導出部102の出力である最適システム構成パラメータを読み込み、追加するサブシステムの目標数を決定する（ステップS1501）。

#### 【0197】

(2) ステップS1501で読み込んだ最適システム構成パラメータと現在のシステム構成パラメータとを比較し、現在のシステム構成パラメータで性能が不足しているか否かを判定し、不足していなかった場合、後述するステップS1511へ移行する（ステップS1502）。

#### 【0198】

(3) ステップS1502の判定で、現在のシステム構成パラメータで性能が不足していた場合、全ての使用中でない、無料で使用可能な予備のサブシステムを選択し、そのサブシステムを現在のシステム構成に追加する。サブシステム追加の処理の詳細は、図18を用いて後述する（ステップS1503、S1504）。

#### 【0199】

(4) サブシステムの追加によるサブシステム数が目標値に達し、現在のシステ

ム構成が最適システム構成となったか否かを判定し、目標値に達していれば、ここでの処理を終了する（ステップ S 1 5 0 5）。

#### 【0 2 0 0】

（5）ステップ S 1 5 0 5 の判定で、目標値に達していなかった場合、全ての使用中無料予備サブシステムの出力装置に対して、使用許可要求を表示させ、所有者等に使用許可を問い合わせるメッセージを送信する（ステップ S 1 5 0 6）。

#### 【0 2 0 1】

（6）使用許可の問い合わせに対して、使用中無料予備システムの所有者等からサブシステムの使用許可があったか否かを判定し、使用許可が入力された場合、サブシステムの追加処理を行う（ステップ S 1 5 0 7、S 1 5 0 8）。

#### 【0 2 0 2】

（7）サブシステムの追加によるサブシステム数が目標値に達し、現在のシステム構成が最適システム構成となったか否かを判定し、目標値に達していれば、ここでの処理を終了する（ステップ S 1 5 0 9）。

#### 【0 2 0 3】

（8）ステップ S 1 5 0 9 の判定で、目標値に達していなかった場合、または、ステップ S 1 5 0 7 の判定で、使用許可が得られなかった場合、ステップ S 1 5 0 6 で問合せメッセージを送信してから一定の時間が経過したか否か、または、全ての問合せに対して返答があったか否かを判定し、一定の時間が経過していなかった場合、または、問合せに対してまだ全ての返答がなかった場合、ステップ S 1 5 0 7 からの処理に戻って処理を繰り返す。また、一定の時間が経過し、かつ、全ての問合せに対して返答があった場合、後述するステップ S 1 5 1 8 からの処理に移行する（ステップ S 1 5 1 0）。

#### 【0 2 0 4】

（9）ステップ S 1 5 1 1 の判定で、現在のシステム構成パラメータで性能が不足していなかった場合、現在のシステム構成が過剰性能であるか否かを判定し過剰性能でなければ、ここでの処理を終了する（ステップ S 1 5 1 1）。

#### 【0 2 0 5】

（10）現在システム構成で使用されていない全ての有料サブシステムについて、

ステップ S 1 5 1 3 及び S 1 5 1 4 の処理を実行することを設定し、有料サブシステムの削除処理を行う。この削除処理の詳細は、図 1 9 を用いて後に詳述する（ステップ S 1 5 1 2、S 1 5 1 3）。

#### 【0 2 0 6】

(11) サブシステムの削除によりサブシステム数が目標値に達し、現在のシステム構成が最適システム構成となったか否かを判定し、目標値に達していれば、ここでの処理を終了する（ステップ S 1 5 1 4）。

#### 【0 2 0 7】

(12) 全ての有料サブシステムに対する処理の後、ステップ S 1 5 1 4 の判定で、目標値に達していなかった場合、現在システム構成で使用されていない全ての無料サブシステムについて、ステップ S 1 5 1 6 及び S 1 5 1 7 の処理を実行することを設定し、無料サブシステムの削除処理を行う（ステップ S 1 5 1 5、S 1 5 1 6）。

#### 【0 2 0 8】

(13) サブシステムの削除によりサブシステム数が目標値に達し、現在のシステム構成が最適システム構成となったか否かを判定し、目標値に達していれば、ここでの処理を終了する。また、全ての無料サブシステムに対する処理の後も、目標値に達していなかった場合にも、ここでの処理を終了する（ステップ S 1 5 1 7）。

#### 【0 2 0 9】

(14) ステップ 1 5 1 0 の判定で、一定の時間が経過し、かつ、全ての問合せに対して返答があった場合、全ての使用中でない有料予備サブシステムについて、ステップ S 1 5 1 9 とステップ S 1 5 2 0 との処理を実行することを設定し、有料サブシステムの追加処理を行う（ステップ S 1 5 1 8、S 1 5 1 9）。

#### 【0 2 1 0】

(15) サブシステム数が目標値に達し、現在のシステム構成が最適システム構成となったか否かを判定し、目標値に達していれば、ここでの処理を終了する（ステップ S 1 5 2 0）。

#### 【0 2 1 1】

(16) ステップ S 1 5 2 0 の判定で、目標値に達していなかった場合、全ての使用中有料予備サブシステムの出力装置に対して、使用許可要求を表示させ、所有者等に使用許可を問い合わせるメッセージを送信する（ステップ S 1 5 2 1）。

#### 【0 2 1 2】

(17) 使用許可の問い合わせに対して、使用中有料予備システムの所有者等からサブシステムの使用許可があったか否かを判定し、使用許可が入力された場合、サブシステムの追加処理を行う（ステップ S 1 5 2 2、S 1 5 2 3）。

#### 【0 2 1 3】

(18) サブシステムの追加によるサブシステム数が目標値に達し、現在のシステム構成が最適システム構成となったか否かを判定し、目標値に達していれば、ここでの処理を終了する（ステップ S 1 5 2 4）。

#### 【0 2 1 4】

(19) ステップ S 1 5 2 4 の判定で、目標値に達していなかった場合、または、ステップ S 1 5 2 2 の判定で、使用許可が得られなかった場合、ステップ S 1 5 2 1 で問合せメッセージを送信してから一定の時間が経過したか否か、または、全ての問合せに対して返答があったか否かを判定し、一定の時間が経過していなかった場合、または、問合せに対してまだ全ての返答がなかった場合、ステップ S 1 5 2 2 からの処理に戻って処理を繰り返す。また、一定の時間が経過し、かつ、全ての問合せに対して返答があった場合、処理を終了する（ステップ S 1 5 2 5）。

#### 【0 2 1 5】

図 1 8 は前述で説明したフローにおけるサブシステム追加の処理を説明するタイミングチャートであり、次に、これについて説明する。

#### 【0 2 1 6】

(1) 構成変更指示部 1 4 0 3 は、予備サブシステムにジョブ実行初期化指令を送信すると共に、同時実行処理数を送信する（ステップ 1 8 0 2、1 8 0 3）。

#### 【0 2 1 7】

(2) 予備サブシステムは、構成変更指示部からのステップ 1 8 0 2、1 8 0 3 の処理で送られたデータを受信すると、現在実行中の処理が全て終了すると直ぐ

に、トランザクション処理を行うための初期化処理を行い、処理要求の待ち状態になる（ステップ 1 8 0 8 ～ 1 8 1 1）。

#### 【 0 2 1 8 】

（３）予備サブシステムは、自サブシステムの初期化処理が完了すると、構成変更指示部に、トランザクション処理初期化完了通知を送信する（ステップ 1 8 1 2）。

#### 【 0 2 1 9 】

（４）構成変更指示部は、トランザクション処理初期化完了通知を受信すると、性能保証対象コンピュータが有するトランザクション分配部 1 4 0 8 にトランザクション分配開始指令を送信する（ステップ 1 8 0 4、1 8 0 5）。

#### 【 0 2 2 0 】

（５）トランザクション分配部 1 4 0 8 は、トランザクション分配開始指令を受信すると、追加対象の予備サブシステムを、サブシステムリストへ追加することにより、トランザクション分配部がトランザクション処理要求の分配を開始し、構成変更指示部にトランザクション分配開始通知を送信する（ステップ 1 8 1 3 ～ 1 8 1 6）。

#### 【 0 2 2 1 】

（６）構成変更指示部 1 4 0 3 は、トランザクション分配開始通知を受信すると、システム構成データを更新する（ステップ 1 8 0 6、1 8 0 7）。

#### 【 0 2 2 2 】

図 1 9 は図 1 6 のフローで説明したサブシステム削除の処理を説明するタイミングチャートであり、次に、これについて説明する。

#### 【 0 2 2 3 】

（１）まず、構成変更指示部は、サブシステム削除処理において、トランザクション分配終了指令をトランザクション分配部に送信する（ステップ 1 9 0 2）。

#### 【 0 2 2 4 】

（２）トランザクション分配部は、トランザクション分配終了指令を受信し、サブシステムリストから削除対象のサブシステムを削除して、削除処理完了通知を構成変更指示部に送信する（ステップ 1 9 1 1 ～ 1 9 1 3）。

**【0225】**

(3) 構成変更指示部は、トランザクション分配部からの削除処理完了通知を受信することにより、トランザクション分配の終了を確認し、サブシステムにトランザクション処理終了指令を送信する（ステップ1903、1904）。

**【0226】**

(4) サブシステムは、構成変更指示部のトランザクション処理終了指令を受信すると、全ての処理中のジョブを終了させ、トランザクション処理の終了通知を構成変更指示部に送信して、トランザクション処理の終了処理を行う（ステップ1907～1909）。

**【0227】**

(5) 構成変更指示部は、トランザクション処理の終了通知を受信すると、システム構成データを更新する（ステップ1905、1906）。

**【0228】**

前述で説明した構成変更指示部におけるサブシステム追加処理（図18のステップ1801）及びサブシステム削除処理（図19のステップ1901）は、サブシステム、例えば、図14に示すサブシステム1409、または、トランザクション分配部1408とのデータの送受信により、性能保証対象コンピュータのシステム構成を動的に変更させている。

**【0229】**

前述した本発明の第4の実施形態において、有料サブシステム及び無料サブシステムは、使用されていないとき、無断で使用可能なサブシステムと使用できないサブシステムに分類されている。そして、性能保証対象コンピュータの処理能力が不足する場合、応答時間がA以上である確率をB以下にするため、ジョブ処理用のサブシステムを増やす必要があり、その場合、その必要なサブシステムの数を、すでに説明した方法で予測し、次に説明する手順で予備サブシステムを使用する。

**【0230】**

予備サブシステム群に使用許可を得るために、その旨のメッセージを送る。

各予備サブシステムは、メッセージを受信し、出力装置にサブシステムの利用

の可否を問うメッセージを出力して使用者に知らせる。

使用者からの許可が得られた場合、予備サブシステムをジョブ処理用サブシステムに変更しジョブ実行させる。

予備サブシステムの所有者が、サブシステムの所有者と異なる場合、単位時間当たりのサブシステム使用量を使用者に通知する。

予備サブシステムの総使用量に相当する金額を送金する。

#### 【 0 2 3 1 】

一般に、性能保証対象コンピュータ 1 1 0 に対して設定されるトランザクションの応答時間 A と、トランザクションの処理時間が応答時間 A を越える確率 B について、応答時間が A 以上である確率が B 以下であるシステム構成は、C P U の処理性能、C P U 数、主記憶容量等の組み合わせとして多数存在している。

#### 【 0 2 3 2 】

前述した本発明の実施形態による各処理は、処理プログラムとして構成することができ、この処理プログラムは、H D、D A T、F D、M O、D V D - R O M、C D - R O M 等の記録媒体に格納して提供することができる。

#### 【 0 2 3 3 】

また、前述した本発明の実施形態は、与えられた応答時間 A と確率 B とに対して、応答時間が A 以上である確率が B 以下となるシステム構成のうち、最も安価なシステム構成を導出するとして説明したが、本発明は、与えられた応答時間 C と確率 D とに対して、応答時間が C 以下である確率が D 以上となるシステム構成のうち、最も安価なシステム構成を導出するようにコンピュータすることもでき、また、与えられた応答時間値 A、確率 B 及びジョブの処理要求の到着率 E に対して、応答時間が A 以上である確率が B 以下であるシステム構成のうち、最も安価なシステム構成を導出するようにすることもできる。

#### 【 0 2 3 4 】

前述した本発明の各実施形態は、一定の処理性能を保証するシステム構成で、価格が最小となるものを選び出すことができ、これにより、必要な処理性能を持つコンピュータシステムで、価格が最小となるものを自動的かつ短時間で決定することができる。

**【 0 2 3 5 】**

この結果、前記した本発明の実施形態によれば、使用者にとって必要なシステム構成であって、最も低価格のシステム構成がただ 1 つだけ決定されるため、システム構成決定のために不必要な時間と労力を費やす必要がなくなるという効果を得ることができる。

**【 0 2 3 6 】****【発明の効果】**

以上説明したように本発明によれば、必要な処理性能を持つコンピュータのシステム構成の中の価格が最小となるものを自動的かつ短時間で決定することができる。

**【図面の簡単な説明】****【図 1】**

本発明の第 1 の実施形態によるコンピュータのシステム構成導出装置を含む構成導出システムの構成を示すブロック図である。

**【図 2】**

ジョブデータ記憶装置 1 0 7 に格納されているジョブデータの構成を説明する図である。

**【図 3】**

構成導出部 1 0 2 の処理動作を説明するフローチャートである。

**【図 4】**

図 3 のステップ S 3 0 4 での仮想ページ参照間隔確率分布計算部の処理動作を説明するフローチャートである。

**【図 5】**

図 3 のステップ S 3 0 7 での最適解導出の処理動作を説明するフローチャートである。

**【図 6】**

システム構成記憶装置に格納されているシステム構成データの構成を説明する図である。

**【図 7】**

価格データ記憶装置に格納されている価格データの構成を説明する図である。

【図 8】

仮想ページ参照間隔確率の確率分布の例を示す図である。

【図 9】

性能と価格との関係を表すグラフの例を示す図である。

【図 1 0】

本発明の第 2 の実施形態での構成導出部の処理動作を説明するフローチャート（その 1）である。

【図 1 1】

本発明の第 2 の実施形態での構成導出部の処理動作を説明するフローチャート（その 2）である。

【図 1 2】

図 1 1 におけるステップ S 1 0 0 8 での同時実行ジョブ数計算の処理を説明するフローチャートである。

【図 1 3】

性能保証対象コンピュータが予備システム構成要素を有する場合における本発明の第 3 の実施の形態の構成を示すブロック図である。

【図 1 4】

1 つの性能保証対象コンピュータが、複数のサブシステムからなる場合の本発明の第 4 の実施形態の構成を示すブロック図である。

【図 1 5】

本発明の第 4 の実施形態におけるシステム変更指示部の処理動作を説明するフローチャート（その 1）である。

【図 1 6】

本発明の第 4 の実施形態におけるシステム変更指示部の処理動作を説明するフローチャート（その 2）である。

【図 1 7】

本発明の第 4 の実施形態におけるシステム変更指示部の処理動作を説明するフローチャート（その 3）である。

**【図 1 8】**

図 1 5 ～ 図 1 7 で説明したフローにおけるサブシステム追加の処理を説明するタイミングチャートである。

**【図 1 9】**

図 1 6 で説明したサブシステム削除の処理を説明するタイミングチャートである。

**【図 2 0】**

サブシステムテーブルの構成を示す図である。実施の形態 4 で使用されるサブシステムテーブルの図である。

**【符号の説明】**

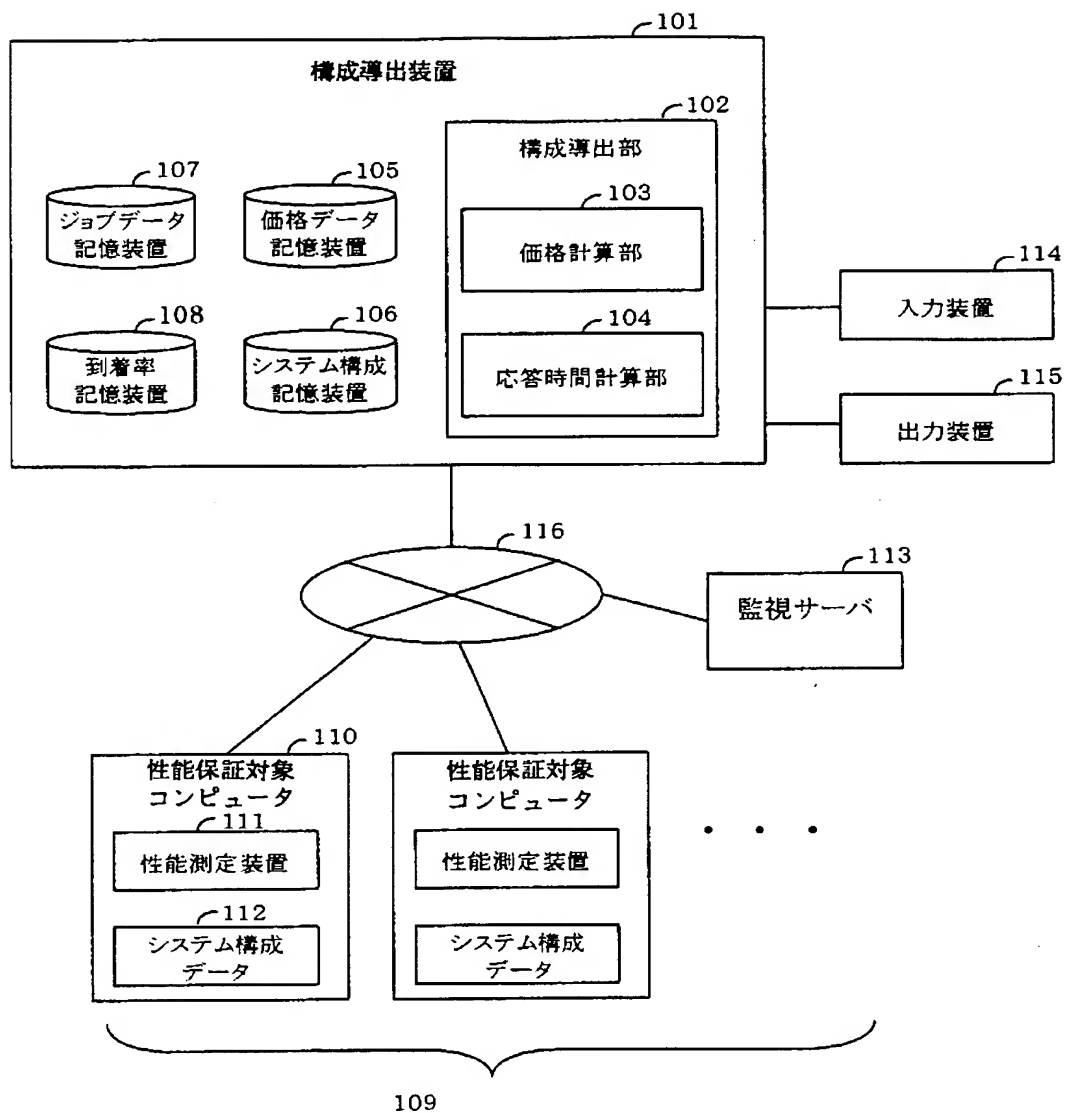
- 1 0 1 構成導出装置
- 1 0 2 構成導出部
- 1 0 3 価格計算部
- 1 0 4 応答時間計算部
- 1 0 5 価格データ記憶装置
- 1 0 6 システム構成記憶装置
- 1 0 7 ジョブデータ記憶装置
- 1 0 8 到着率記憶装置
- 1 0 9 性能保証対象コンピュータ群
- 1 1 0、1 4 0 7 性能保証対象コンピュータ
- 1 1 1 性能測定装置
- 1 1 2 システム構成データ
- 1 1 3 監視サーバ
- 1 1 4、1 4 1 2 入力装置
- 1 1 5、1 4 1 3 出力装置
- 1 1 6 ネットワーク
- 1 3 0 5 性能保証対象コンピュータ
- 1 3 0 6 性能保証対象コンピュータ群
- 1 3 0 7 動的システム構成変更装置

- 1 3 0 8 予備システム構成要素
- 1 4 0 3 構成変更指示部
- 1 4 0 8 トランザクション分配部
- 1 4 0 9 ~ 1 4 1 1 サブシステム 1 ~ 3
- 1 4 1 4 処理結果記憶装置

【書類名】 図面

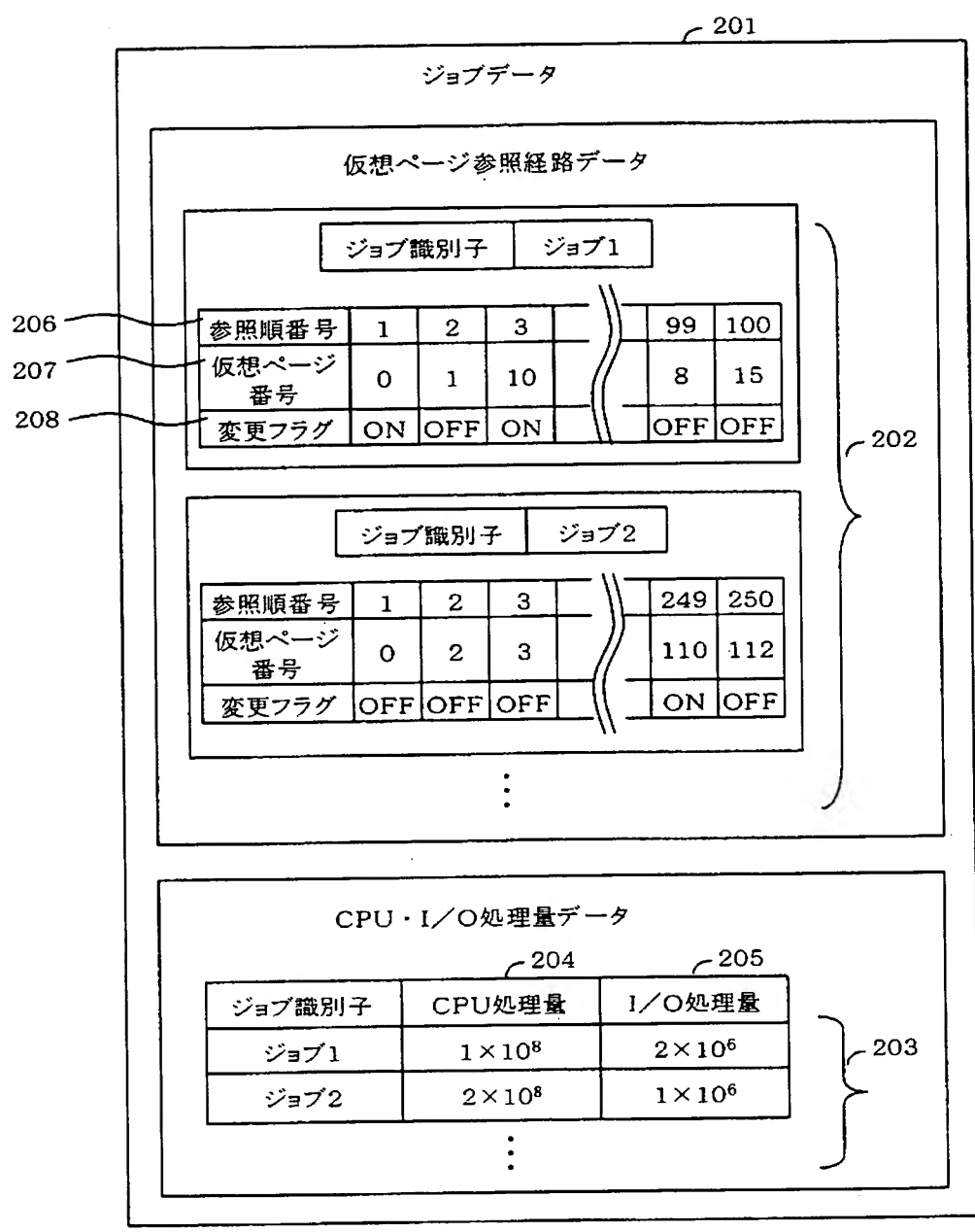
【図 1】

図 1



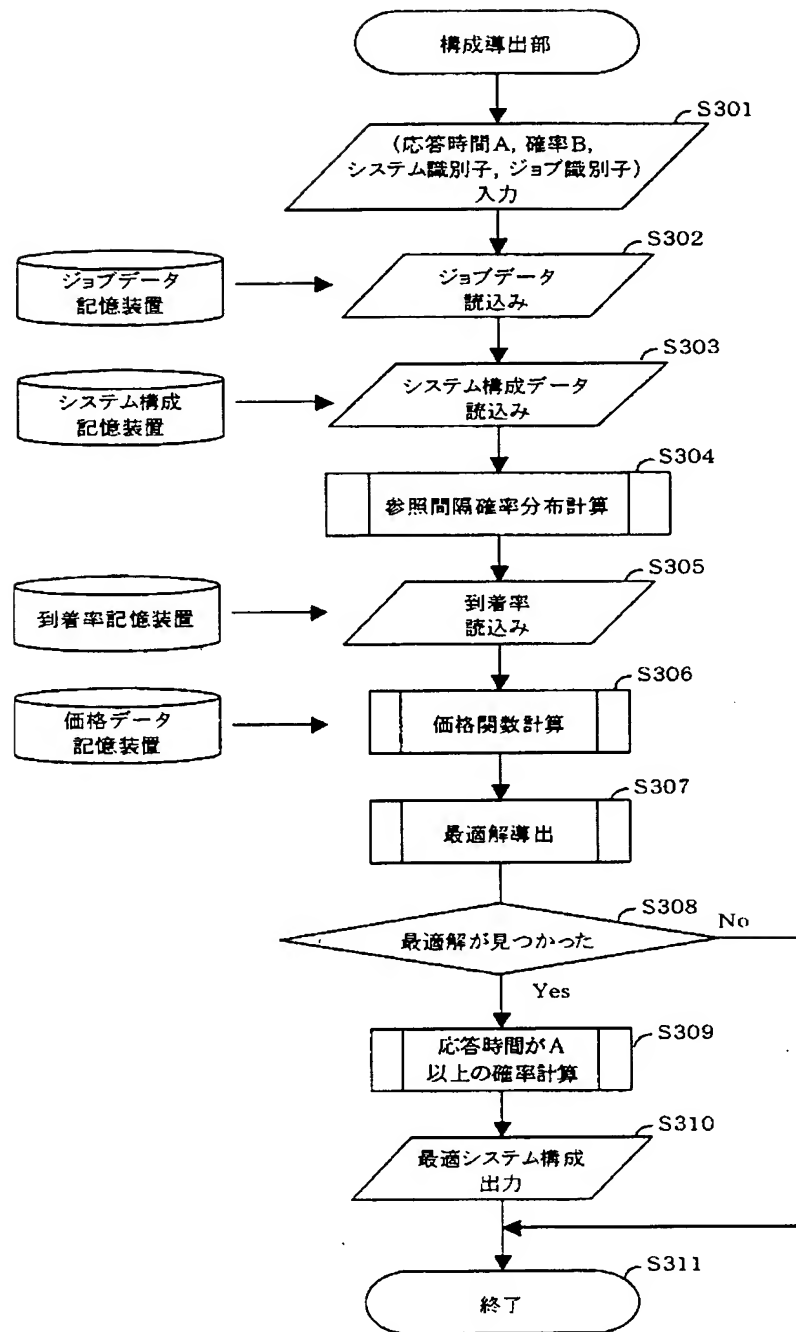
【図 2】

図 2



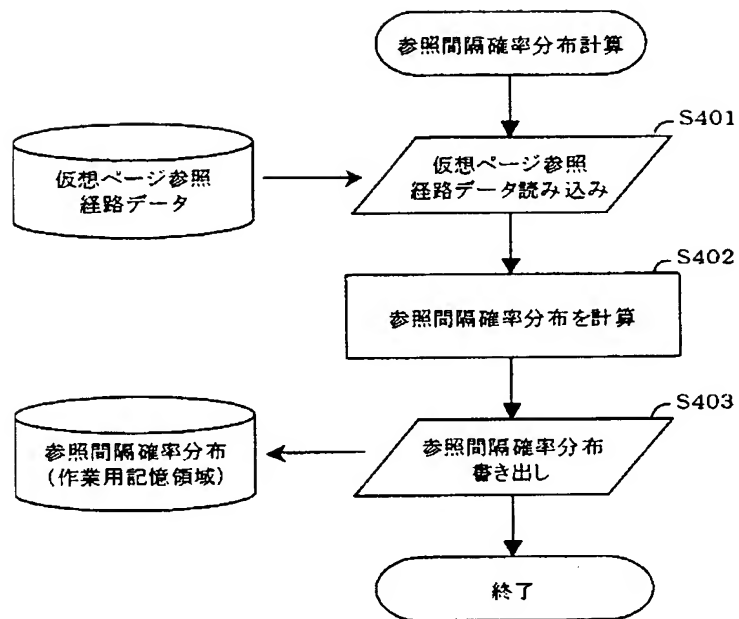
【図 3】

図 3



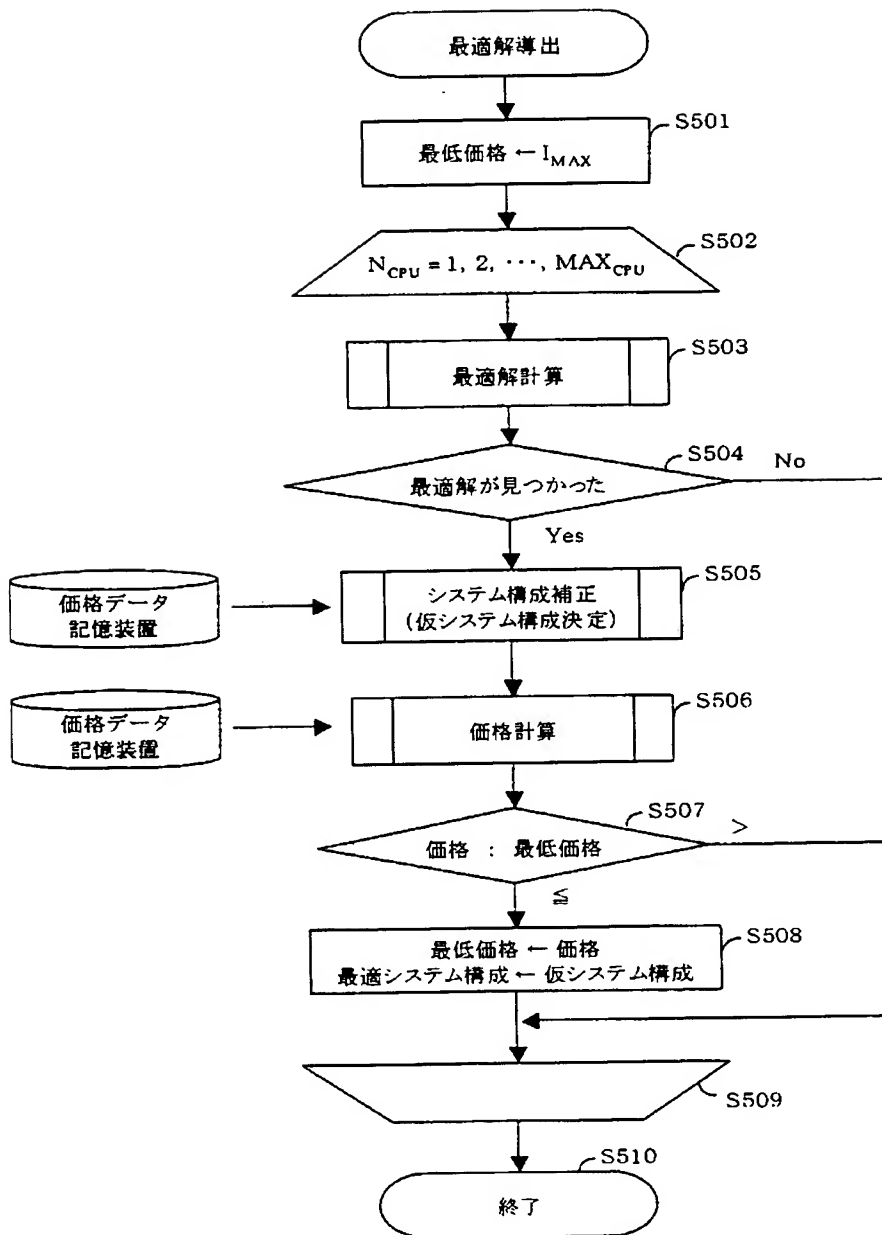
【図 4】

図 4



【図 5】

図 5



【図 6】

図 6

システム構成データ	
システム識別子	システム1
システム構成	性能値
CPU数	2
CPU処理性能(CPU種類)	12000 (MIPS)
主記憶容量	10 (G bytes)
I/O処理速度	100 (M bytes/s)
1ページページング時間	$1.0 \times 10^{-3}$ (s)

601

602

603

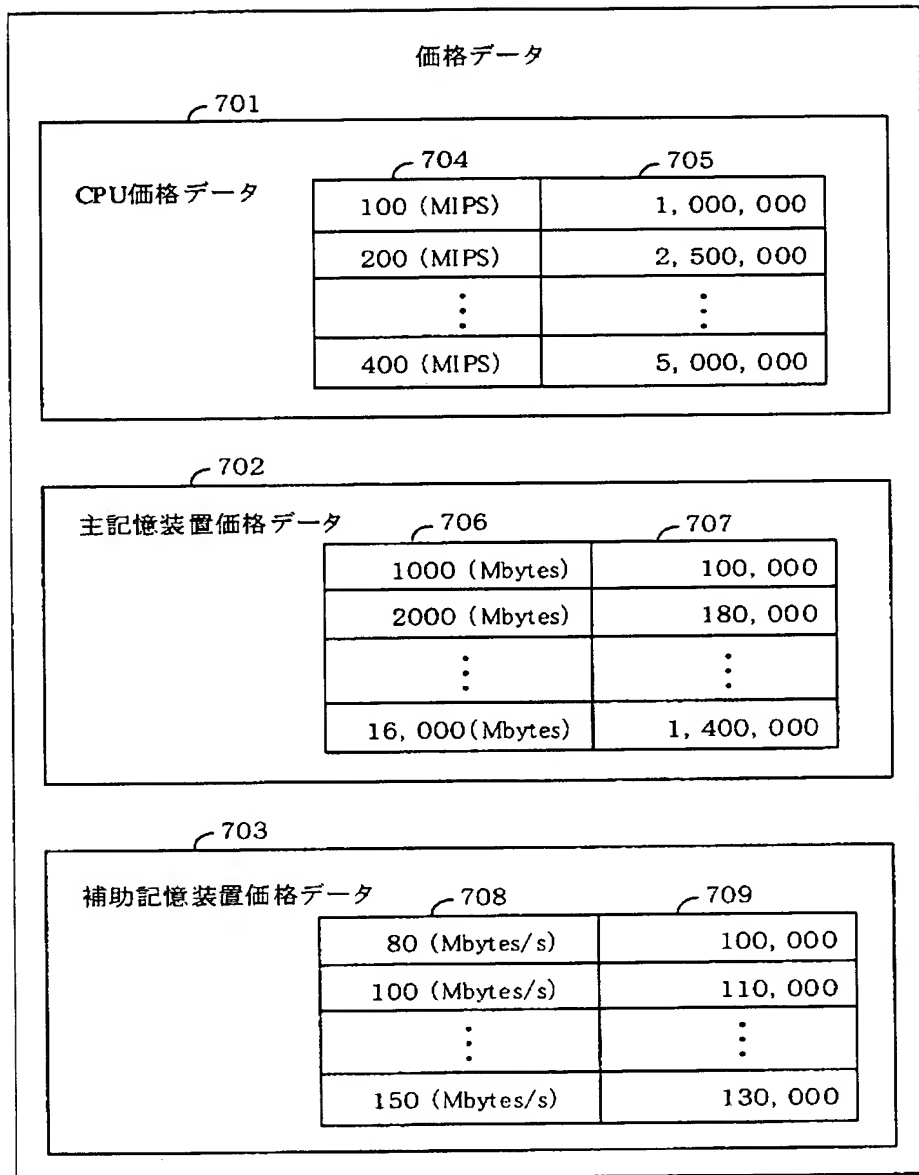
604

605

606

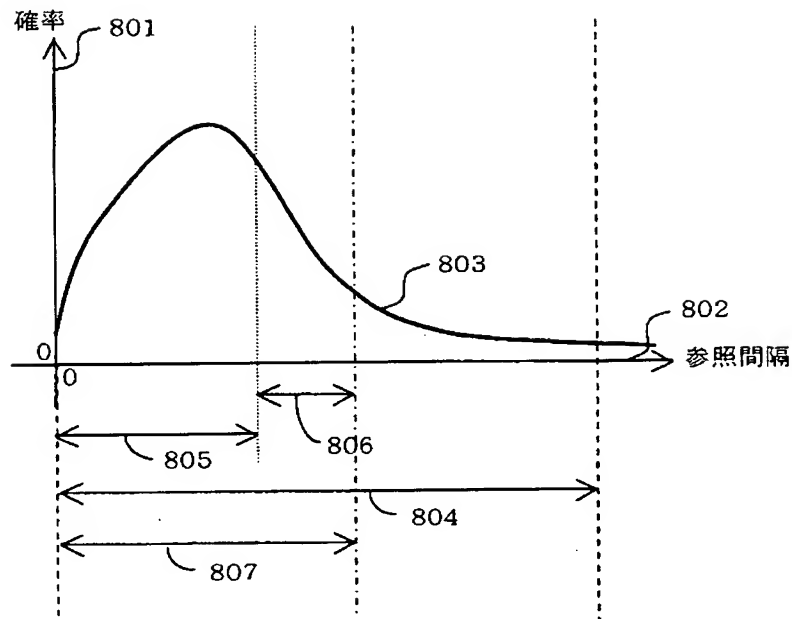
【図 7】

図 7



【図 8】

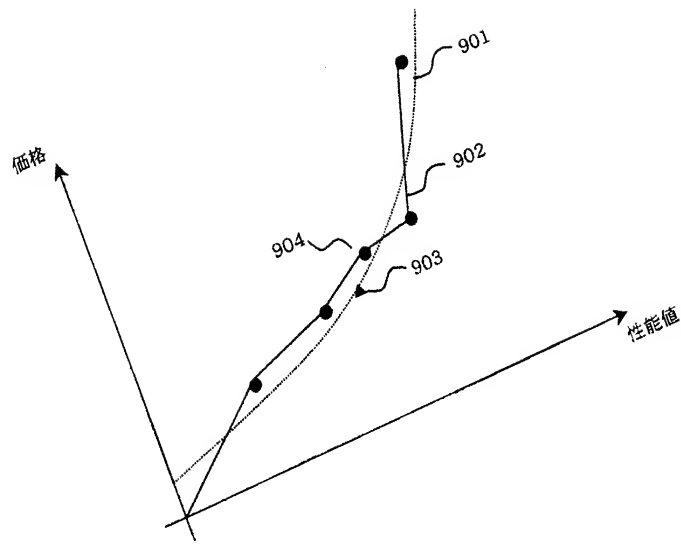
図 8



特願2002-186335

【図9】

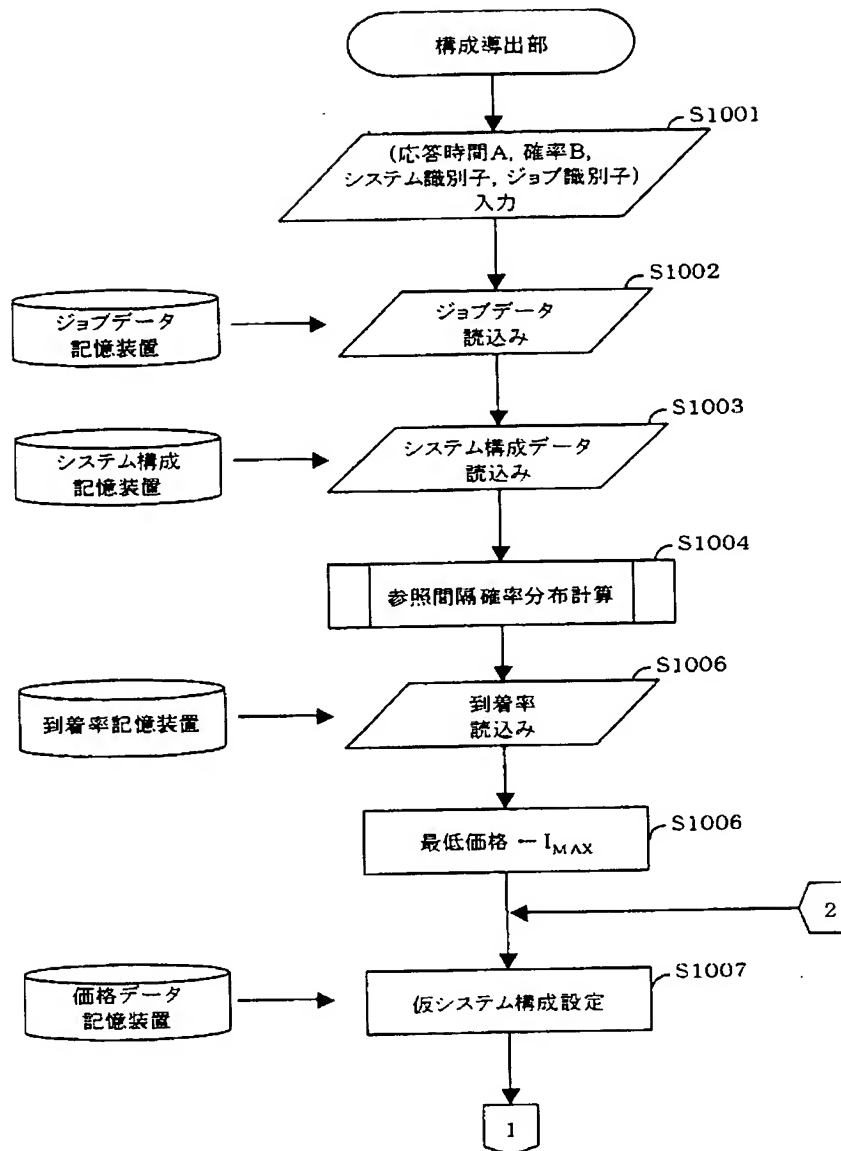
図9



出証特2003-3054348

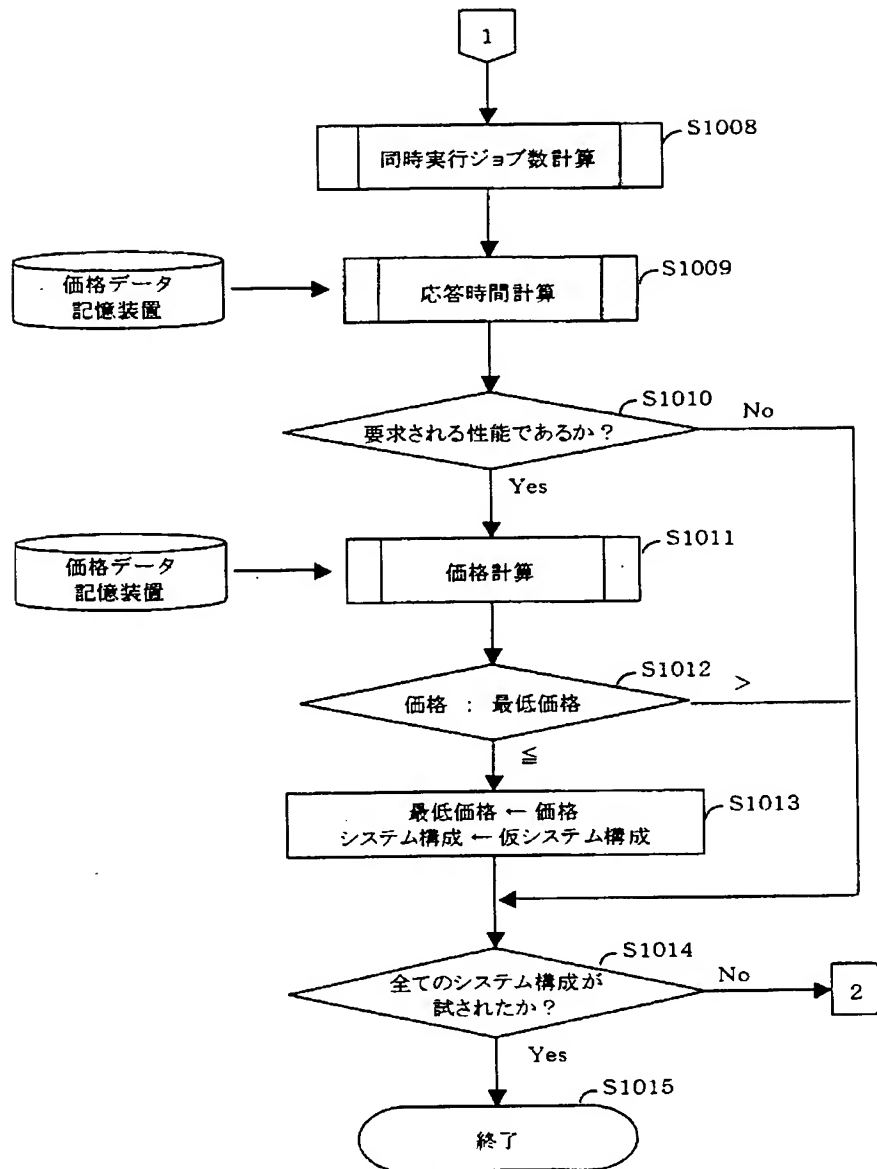
【図10】

図10



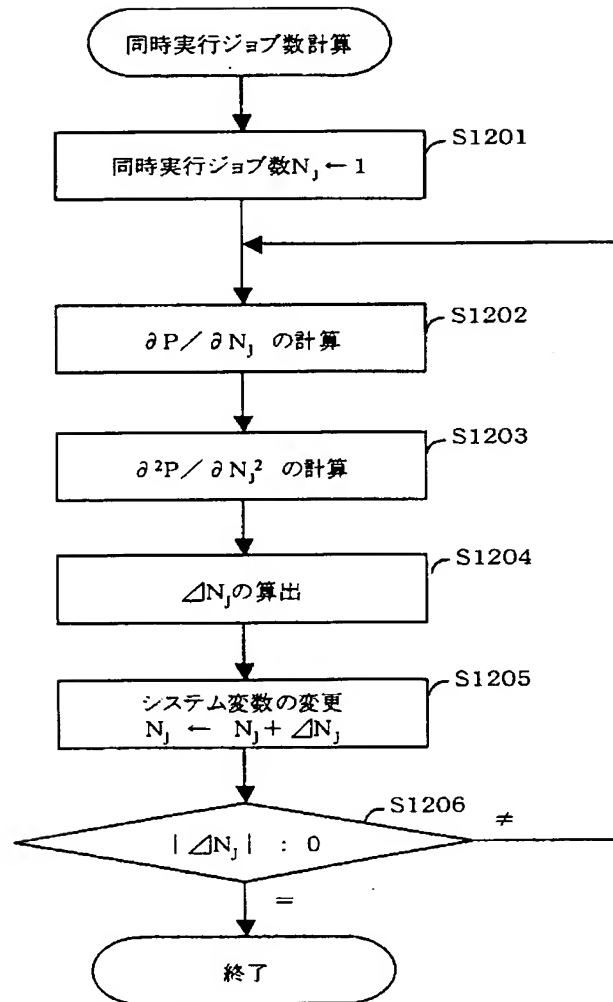
【図 11】

図 11



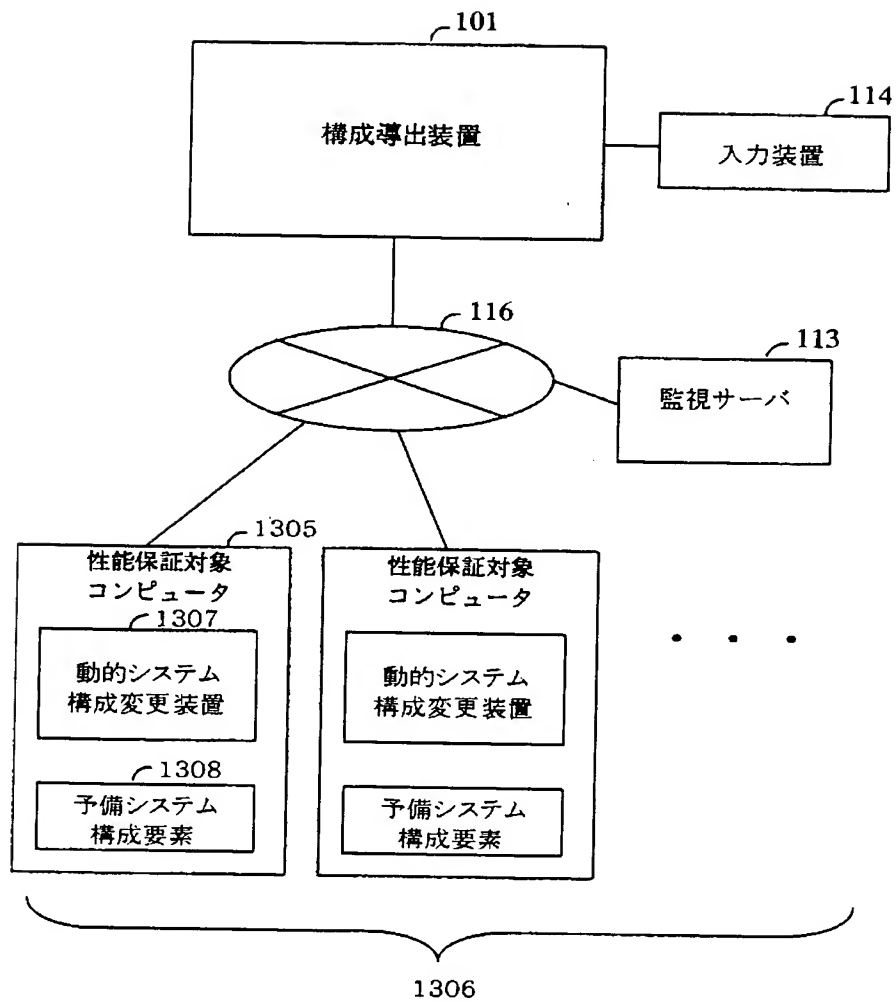
【図 12】

図 12



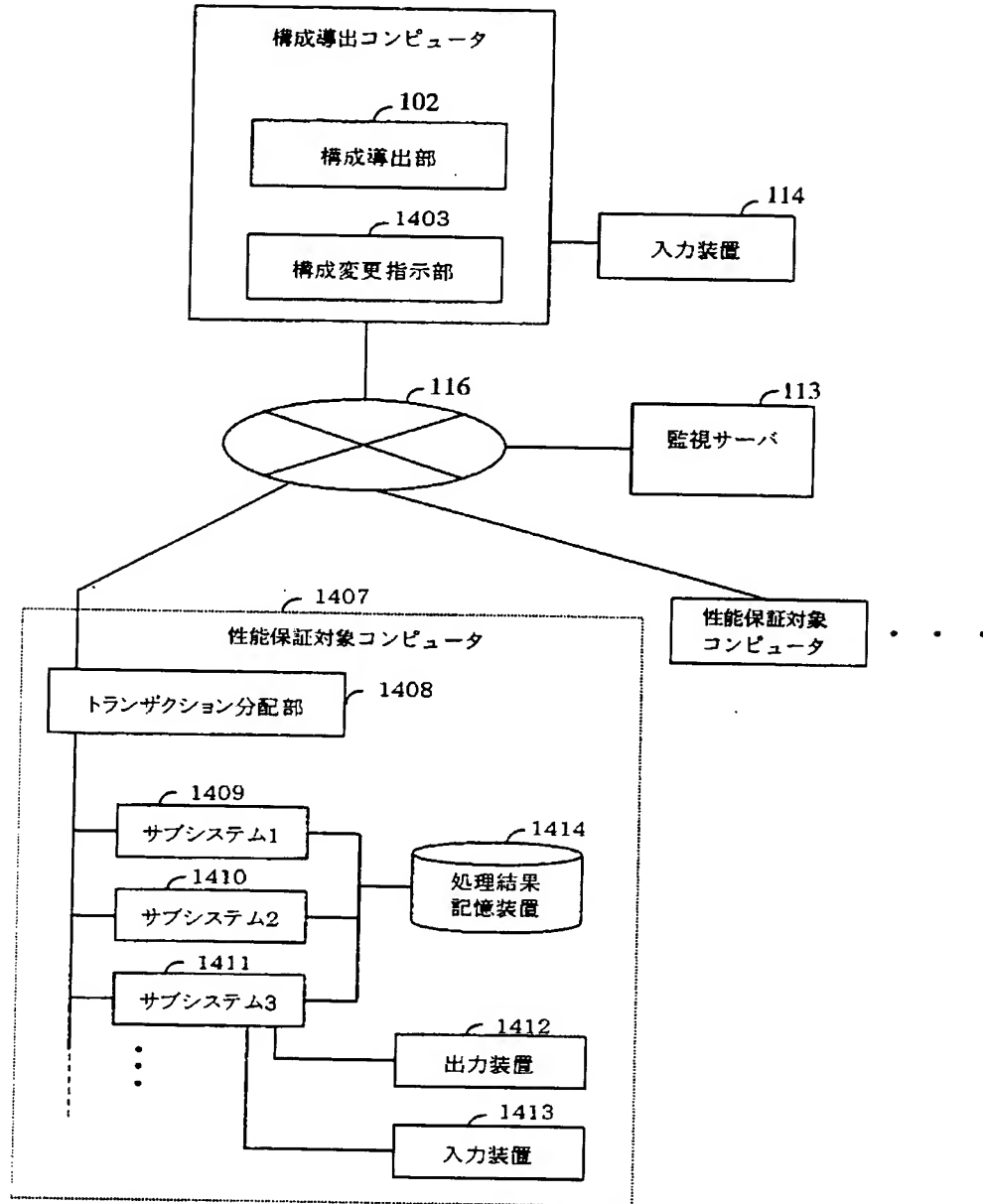
【図 13】

図 13



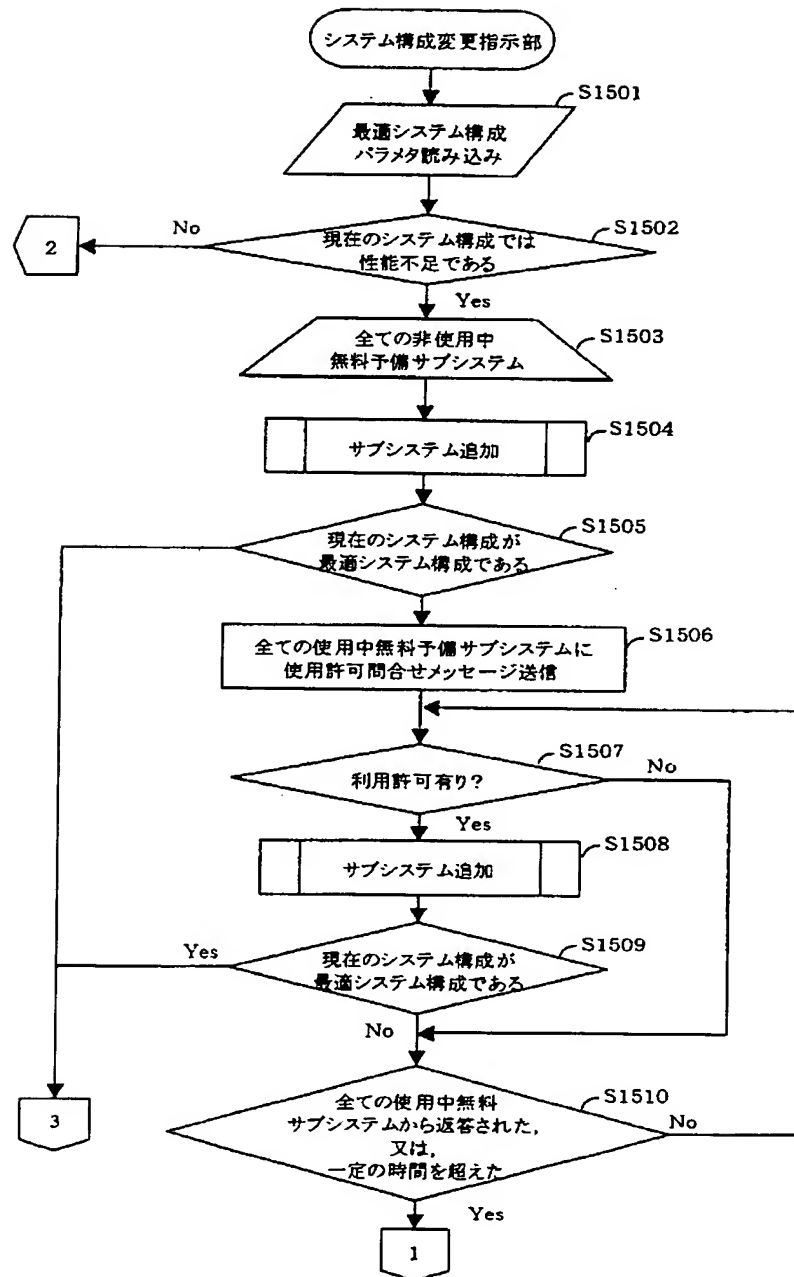
【図 14】

図 14



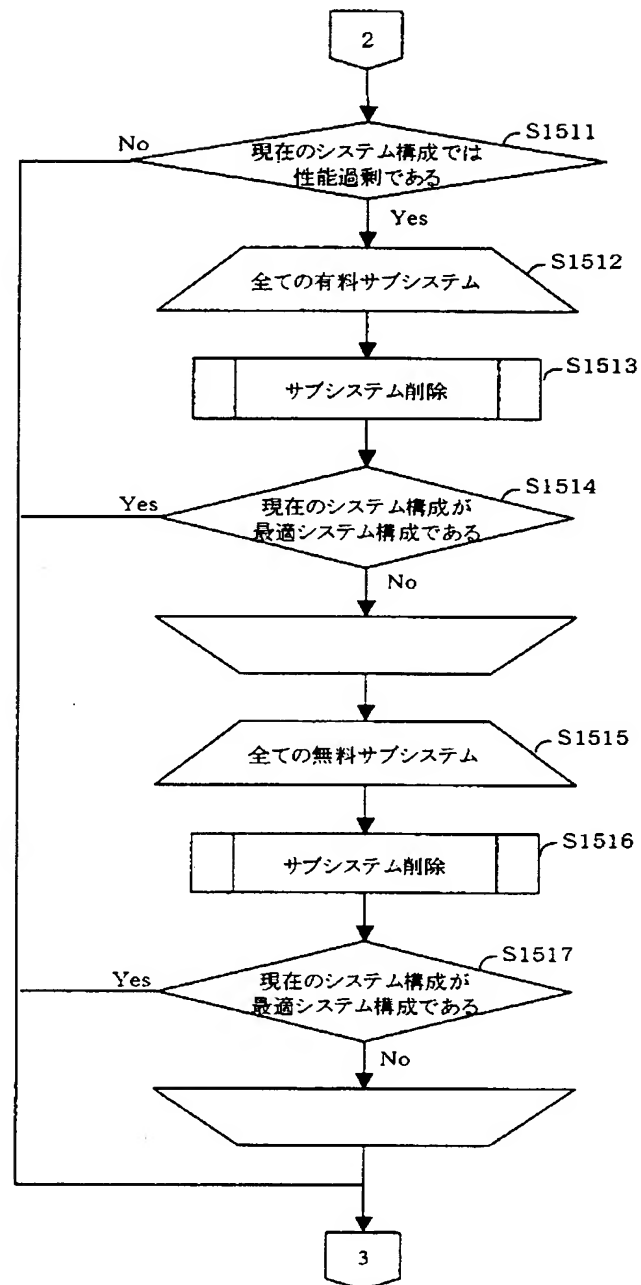
【図 15】

図 15



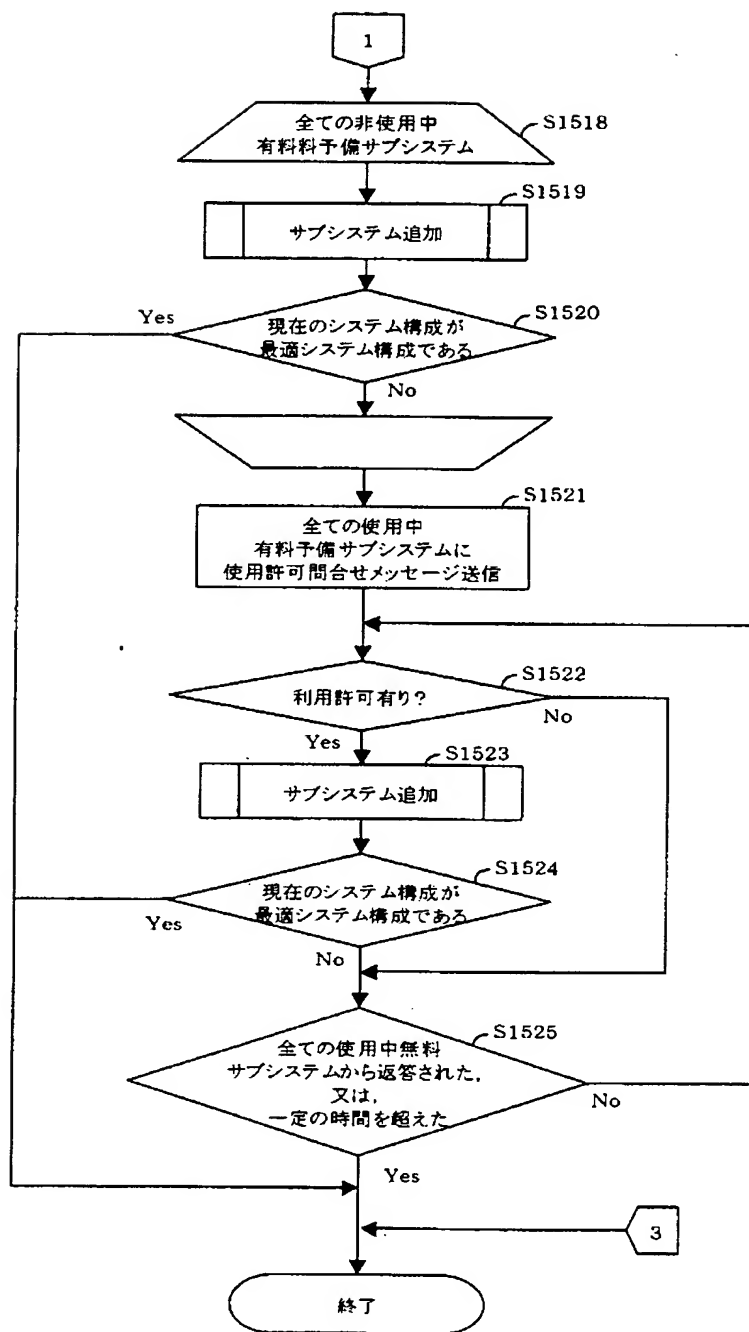
【図 16】

図 16



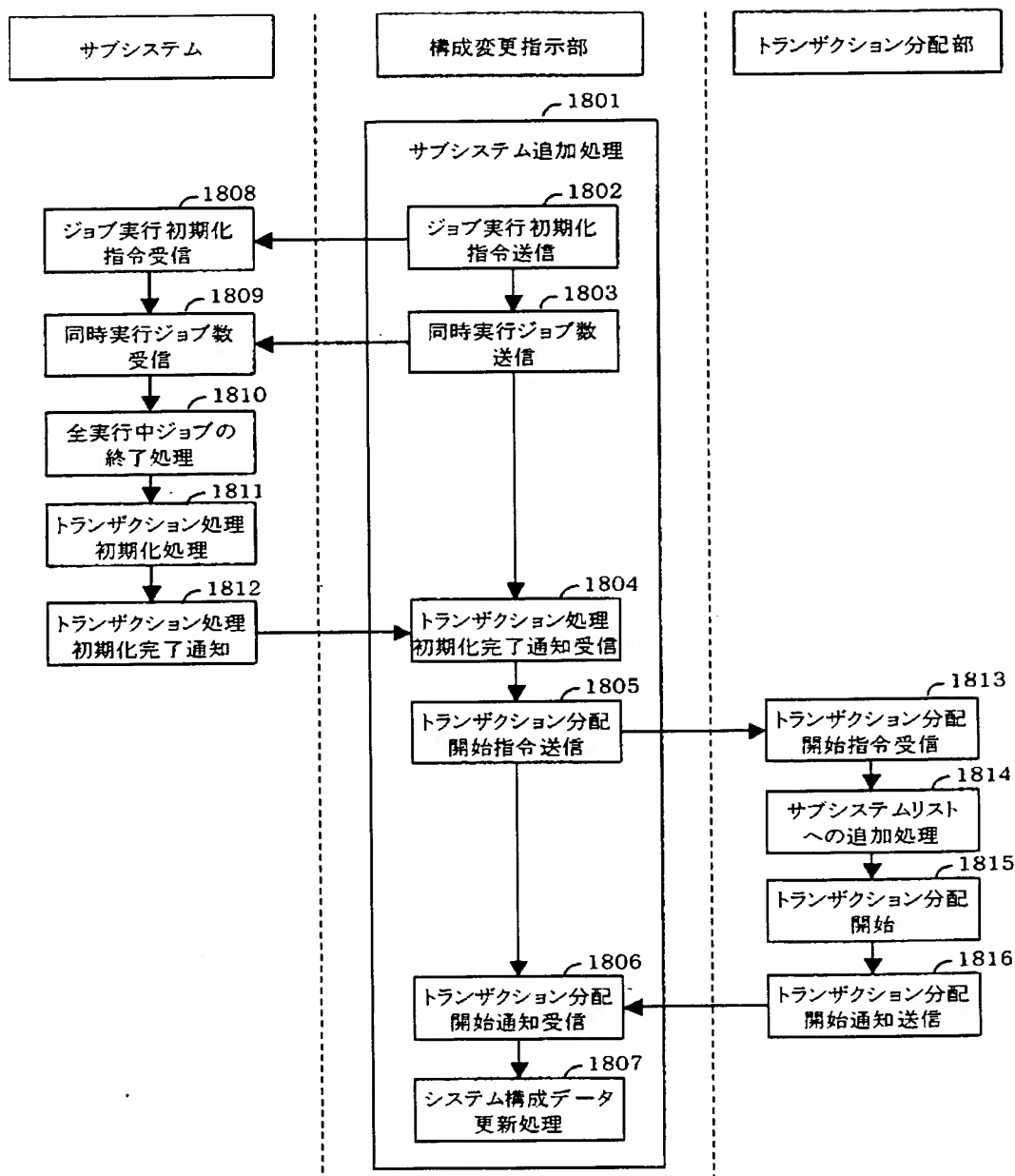
【図 17】

図 17



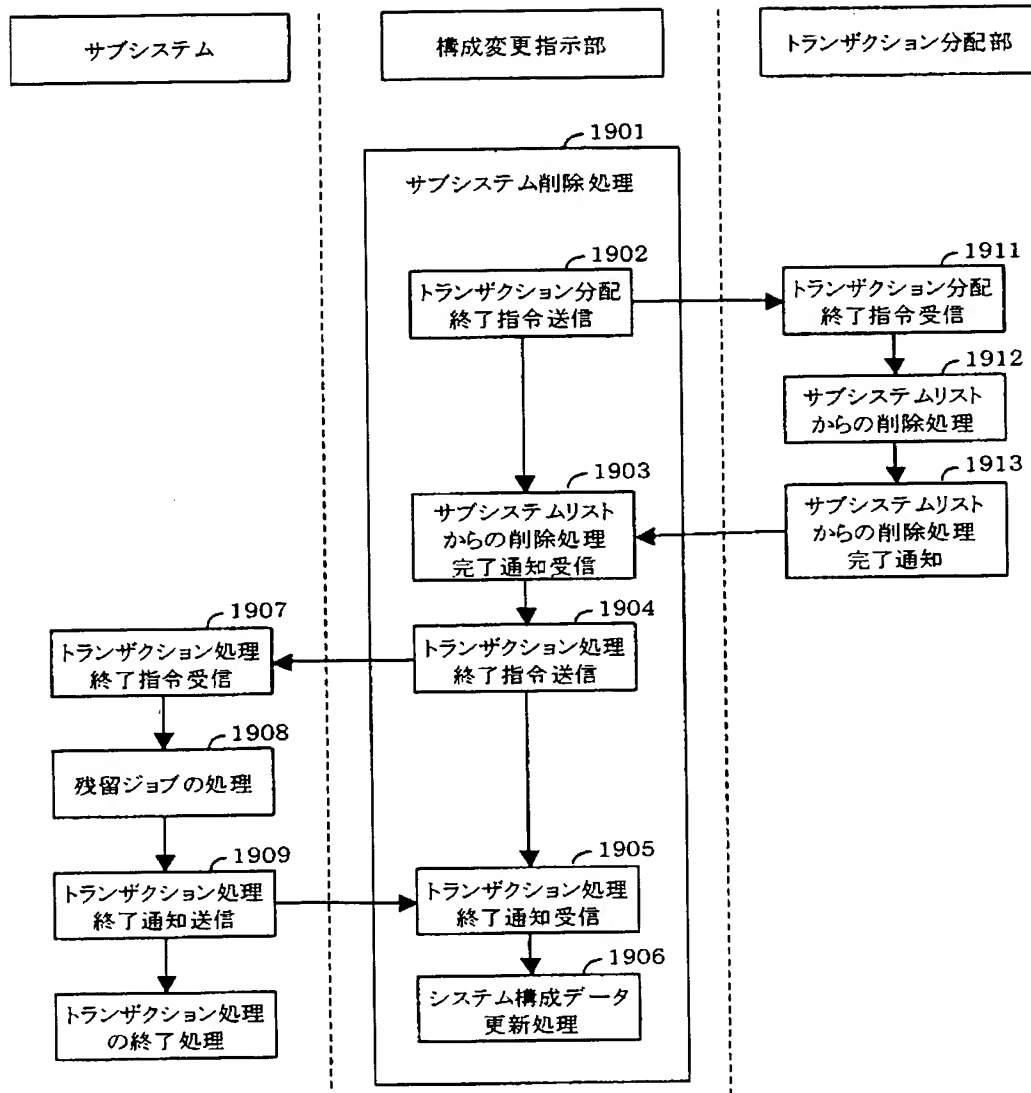
【図 18】

図 18



【図 19】

図 19



【図 20】

図 20

2001 サブシステム識別子	2002 使用中／非使用中フラグ	2003 単位時間使用料(課金額)
001	ON	0
002	OFF	0
003	ON	1, 000
004	ON	2, 000
005	OFF	2, 500

【書類名】 要約書

【要約】

【課題】 必要な処理性能を持つコンピュータのシステム構成の中の価格が最小となるものを自動的にかつ短時間で決定する。

【解決手段】 システム構成を、4つのパラメータ、（CPU性能、CPU性能、主記憶容量、I/O処理速度）で表現し、トランザクション処理の応答時間に関する一定の制約条件のもとで、価格が最小となる4つのパラメータを、ラグランジュの未定乗数法等による数値解法を利用して導出する。一方で、システムが有するCPUの1つが同時に実行できるジョブ数も最適化する。図示例では、応答時間がA以上である確率がB以下であるシステム構成で価格が最小となる構成を導出している。

【選択図】 図3

特願 2 0 0 2 - 1 8 6 3 3 5

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 5 1 0 8 ]

1 . 変更年月日

1 9 9 0 年 8 月 3 1 日

[変更理由]

新規登録

住 所

東京都千代田区神田駿河台 4 丁目 6 番地

氏 名

株式会社日立製作所